

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.

Inter Partes Review No. IPR2020-01582

U.S. Patent No. 8,572,138 B2

PATENT OWNER'S RESPONSE

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND	4
A.	Procedural History.....	4
B.	The '138 Patent	5
III.	LEGAL STANDARDS	6
A.	<i>Inter Partes</i> Review.....	6
B.	Obviousness.....	7
C.	Claim Construction – Order of Steps	8
IV.	PETITIONER'S INVALIDITY GROUNDS FAIL TO ESTABLISH THAT ANY CHALLENGED CLAIM IS UNPATENTABLE.....	10
A.	Petitioner's Alleged Invalidity Grounds	10
B.	Person of Ordinary Skill in the Art	10
C.	Claim Construction – Challenged Claims Require an Ordered Sequence of Steps.....	11
1.	The Specification of the '138 Patent	11
a)	Two-Phase Capture Process	12
b)	Three-Step Deployment Process	22
2.	The Language of the Challenged Claims.....	31
a)	Claim 1.....	32
b)	Claim 19.....	41
c)	Claim 26.....	42
D.	Claim Construction – Petitioner's Proposed Claim Construction Attempts to Read the Deployment Process Out of the Challenged Claims	42
E.	All Challenged Claims Are Valid Over Any Combination of Petitioner's Cited References of Khandekar, Le, Sidebottom, and Stone.....	45
1.	Ground 1: Khandekar.....	45

a)	Overview of Khandekar.....	46
b)	Khandekar Does Not Disclose the Three Step Deployment Process Required by the Challenged Claims	49
c)	Khandekar Fails to Disclose a Number the Elements of the Challenged Claims	57
d)	Advantages of the Challenged Claims over Khandekar.....	61
2.	Ground 2: Khandekar and Le.....	64
3.	Ground 3: Khandekar, Le, and Sidebottom	65
4.	Ground 4: Khandekar, Le, and Stone	65
V.	CONCLUSION.....	66
VI.	CERTIFICATION PURSUANT TO 37 C.F.R. § 42.24(D)	67

TABLE OF AUTHORITIES

Cases

<i>Aristocrat Techs. Australia Pty Ltd. v. Int'l Game Tech.</i> , 709 F.3d 1348, 1356-57 (Fed. Cir. 2013)	45
<i>Custom Accessories, Inc. v. Jeffrey-Allan Indus., Inc.</i> , 807 F.2d 955, 962 (Fed. Cir. 1986)	7
<i>Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.</i> , 800 F.3d 1375, 1378 (Fed. Cir. 2015)	6
<i>Function Media, LLC v. Google, Inc.</i> , 708 F.3d 1310, 1320 (Fed. Cir. 2013)	8
<i>In re Gartside</i> , 203 F.3d 1305 (Fed. Cir. 2000)	7
<i>In re Kubin</i> , 561 F.3d 1351, 1359-60 (Fed. Cir. 2009)	8
<i>In re Lee</i> , 277 F.3d 1338, 1343 (Fed. Cir. 2002)	7
<i>KSR Int'l Co. v. Teleflex Inc.</i> , 550 U.S. 398, 401 (2007)	7
<i>Leo Pharm. Prods. Ltd. v. Rea</i> , 726 F.3d 1346, 1356 (Fed. Cir. 2013)	8
<i>Mformation Techs., Inc. v. Rsch. in Motion Ltd.</i> , 764 F.3d 1392, 1398-400 (Fed. Cir. 2014)	8, 11, 44
<i>Panduit Corp. v. Dennison Mfg. Co.</i> , 810 F.2d 1561, 1565-68 (Fed. Cir. 1987)	7
<i>Process Control Corp. v. HydReclaim Corp.</i> , 190 F.3d 1350, 1356-57 (Fed. Cir. 1999)	40

SIBIA Neurosciences, Inc. v. Cadus Pharm. Corp.,
225 F.3d 1349, 1355 (Fed. Cir. 2000)7

Teleflex, Inc. v. Ficosa N. Am. Corp.,
299 F.3d 1313, 1334 (Fed. Cir. 2002)8

Statutes

35 U.S.C. § 103 46, 64, 65

35 U.S.C. § 316(e) 1, 6, 66

LIST OF EXHIBITS

1001	U.S. Patent No. 8,572,138 ("the '138 Patent")
1002	Prosecution File History for the '138 Patent
1003	Declaration of Prashant Shenoy
1004	U.S. Patent No. 7,577,722 to Khandekar et al. ("Khandekar")
1005	U.S. Patent No. 8,209,680 to Le et al. ("Le")
1006	U.S. Patent No. 7,962,633 to Sidebottom et al. ("Sidebottom")
1007	U.S. Patent Application Publication No. 2003/0036886 to Stone ("Stone")
1008	Khanna et al., <i>Application Performance Management in Virtualized Server Environments</i> , NOMS 2006 ("Khanna")
1009	Urgaonkar et al., <i>Dynamic Provisioning of Multi-tier Internet Applications</i> , ICAC '05 ("Urgaonkar")
1010	Excerpt of S. Russell, P. Norvig, <i>Artificial Intelligence - A Modern Approach</i> (2d. ed. Prentice-Hall 2003) ("Russell")
1011	Barham et al., <i>Xen and the Art of Virtualization</i> , SOSP '03: ACM Symposium on Operating Systems Principles, 164-177 (Oct. 2003)
1012	U.S. Patent No. 6,496,847 to Bugnion et al. ("Bugnion")
1013	U.S. Patent Application Publication No. 2004/0019672 to Das et al. ("Das")
1014	U.S. Patent Application Publication No. 2007/0220246 to Powell et al. ("Powell")
1015	JP Patent Application Publication No. 2001-318797 to French
1016	English translation of JP Patent Application Publication No. 2001-318797 to French ("French")
1017	Declaration of Sylvia D. Hall-Ellis

1018	Reserved
1019	Reserved
1020	Reserved
1021	Broadcom PR 3-1 and 3-2 Disclosures from <i>Broadcom Corp. et al. v. Netflix, Inc.</i> , Case No. 3:20-cv-04677-JD
1022	Preliminary Infringement Contentions for Sundar '138 <i>Broadcom Corp. et al. v. Netflix, Inc.</i> , Case No. 3:20-cv-04677-JD
2001	Declaration of Dr. David Rosenblum ("Rosenblum Decl.")
2002	Deposition transcript of Dr. Prashant Shenoy ("Shenoy Dep.")
2003	Joint Claim Construction and Prehearing Statement, <i>Broadcom Corp. et al. v. Netflix, Inc.</i> , No. 3:20-cv-04677-JD (N.D. Cal. May 18, 2021), ECF No. 112
2004	Tanenbaum et al., Excerpts from <i>Operating Systems: Design and Implementation</i> , Pearson Education, Inc. (3d. ed. 2006)
2005	Apple, Inc., <i>Mac OS X "Gold Master" Released to Manufacturing</i> , Apple Press Release, Apple Newsroom; available at: https://www.apple.com/newsroom/2001/03/07Mac-OS-X-Gold-Master-Released-To-Manufacturing/ (Mar. 7, 2001)
2006	Excerpt from <i>Information and Communications Technologies: OECD Information Technology Outlook</i> , Organization for Economic Co-Operation and Development, OECD Publishing (2006)

Patent Owner Avago Technologies International Sales Pte. Limited ("Avago" or "Patent Owner") hereby respectfully submits this Response to the Petition for *Inter Partes* Review No. IPR2020-01582 (the "Petition") challenging the patentability of certain claims of U.S. Patent No. 8,572,138 (the "'138 Patent"). This filing is timely under the Board's March 15, 2021 Scheduling Order (Paper 10) and the Parties' Joint Notice of Stipulation to Change Due Dates (Paper 12).

I. INTRODUCTION

Netflix, Inc. ("Petitioner") has not proven that any of Claims 1-5, 9, 10, 17, 19-21, and 26 of the '138 Patent (the "Challenged Claims") are unpatentable by a preponderance of the evidence as required by 35 U.S.C. § 316(e) for several independent and distinct reasons.

First, the Challenged Claims contain specific limitations related to storing (by a software repository) and deploying (by a control node): (i) a plurality of image instances of a virtual machine manager and (ii) a plurality of image instances of software applications. As required by the specification of the '138 Patent and the language of the Challenged Claims, and as acknowledged by Petitioner's expert during his deposition, the Challenged Claims require these two sets of image instances to be separately stored and separately deployed. The Challenged Claims further require that, when executed on application nodes, the image instances of the

virtual machine manager provide a plurality of virtual machines on which the image instances of software applications are deployed.

To satisfy these requirements, the Challenged Claims require a specific ordered sequence of steps. Specifically, the Challenged Claims require that the image instances of a virtual machine manager be executed (or be executable) on application nodes to provide virtual machines. The Challenged Claims further require that a plurality of image instances of software applications be executed (or be executable) on the virtual machines provided by the virtual machine manager. As a result, the image instances of the virtual machine manager must be deployed first to provide the virtual machines. Subsequently, the image instances of the software applications are then executed on the plurality of virtual machines – virtual machines that did not exist before being provided by the virtual machine manager. The Challenged Claims (whether the claimed distributed computer system of independent Claim 1, the claimed method of independent Claim 19, or the claimed computer-readable medium of Claim 26) contain this required sequence – a sequence required by the specification and as reflected by the claim language as a matter of both logic and grammar.

Petitioner's prior art does not disclose or suggest these requirements of the Challenged Claims – many of which are clear based on the antecedent basis of the various claim elements such as "a plurality of image instances of a virtual machine

manager," "a plurality of virtual machines," and "a plurality of image instances of software applications." These claim elements are initially defined in each independent claim. Each subsequent reference to these elements refers to these earlier definitions (*e.g.*, when a Challenged Claim recites to "the plurality of virtual machines" and "the virtual machines" in later parts of the Claim, this is a recitation to the same virtual machines defined as "a plurality of virtual machines" earlier in the Claim).

Petitioner's expert acknowledged that he had offered no opinion in his declaration (and declined to offer any opinion during his deposition) with respect to the concept of antecedent basis. For example, Petitioner's expert did not offer any opinion concerning whether the "plurality of virtual machines" provided by the plurality of image instances of the virtual machine manager were the same virtual machines that the software applications must be executable on once deployed. By not constraining his invalidity analysis with the requirements of antecedent basis, Petitioner's expert (and the Petitioner) decided to simply mix and match various pieces of prior art with complete disregard to the required sequence of the Challenged Claims. As a result, Petitioner's prior art falls far short of rendering any of the Challenged Claims obvious under any of the Grounds.

Second, regardless of whether the Board finds that the Challenged Claims contain a required sequence of steps, the Petitioner's prior art still fails to disclose

the elements of the Challenged Claims, including (1) a software image repository to store (i) a plurality of image instances of a virtual machine manager that when executed provide a plurality of "virtual machines" and (ii) a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines, and (2) a control node to provide autonomic deployment of the plurality of image instances of the virtual machine manager **and** the plurality of image instances of the software applications on the virtual machines.

One of the primary shortcomings of the prior art cited by Petitioner is that the term virtual machine (or "VM"), as used in Petitioner's prior art, has a very different meaning than the term "virtual machine," as used in the '138 Patent – an issue that Petitioner ignores, but which permeates throughout its invalidity analysis. When the meaning of the term "VM" in Khandekar and "virtual machine" in the '138 Patent are correctly applied, it is apparent that what Khandekar discloses is very different than what the Challenged Claims require.

II. BACKGROUND

A. Procedural History

Broadcom Corporation ("Broadcom") and Avago filed a lawsuit against Petitioner for infringement of the '138 Patent and eleven other patents. *See Broadcom Corp. et al. v. Netflix, Inc.*, No. 8:20-cv-00529 (C.D. Cal. 2020). The case was subsequently transferred to the N.D. Cal. on July 14, 2020, No. 3:20-cv-

04677 (N.D. Cal. 2020). Petitioner filed this Petition on September 10, 2020 (Paper 2). On December 21, 2020, Patent Owner filed Patent Owner's Preliminary Response to the First Petition (Paper 7). On February 22, 2021, Petitioner filed a second Petition against the additional claims of the '138 Patent ("Second Petition"). *See Netflix, Inc. v. Avago Technologies International Sales PTE. Limited*, IPR2021-00542, Paper 2 (PTAB Feb. 22, 2021). On March 15, 2021, the Board issued its decision granting institution of *inter partes* review of the Challenged Claims in the First Petition ("Institution Decision") (Paper 9). On June 9, 2021, Patent Owner filed Patent Owner's Preliminary Response to the Second Petition. *See Netflix, Inc. v. Avago Technologies International Sales PTE. Limited*, IPR2021-00542, Paper 7 (PTAB June 9, 2021).

B. The '138 Patent

The inventions of the '138 Patent center around a plurality of two sets of images: (1) image instances of a virtual machine manager, and (2) image instances of software applications. *See* Ex. 1001 at 36:25-38. As used in the '138 Patent, the term "virtual machine manager" is a software program that can create or manage one or more virtual machines. *See id.* at 32:60-62, 35:6-9. The term "virtual machine" refers to "software that creates an environment that emulates a computer platform."

Id. at 32:67-33:2, 36:34-36.¹ And, finally, the term "image instance" is a file that contains a snapshot or "image" of an installed and configured software application. *See* Ex. 1001 at 5:48-6:7, 35:1-24; *see also* Ex. 2003 at 2. As discussed in detail below, the Challenged Claims are fundamentally different than – and provide significant advantages over – the prior art relied on in the Petition.

III. LEGAL STANDARDS

A. *Inter Partes* Review

"In an *inter partes* review, the burden of persuasion is on the petitioner to prove 'unpatentability by a preponderance of the evidence,' 35 U.S.C. § 316(e), and that burden never shifts to the patentee." *Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015).

¹ Patent Owner notes that the parties have agreed elsewhere to a construction of "software that emulates a physical computer platform." *See* Ex. 2003 (Joint Claim Construction and Prehearing Statement, *Broadcom Corp. et al. v. Netflix, Inc.*, No. 3:20-cv-04677-JD (N.D. Cal. May 18, 2021), ECF No. 112). For purposes of this *inter partes* review, however, Patent Owner believes there is no material difference between these two constructions.

B. Obviousness

In addition to being novel, a valid claim must not have been obvious to a person with ordinary skill in the art at the time of the invention. *See, e.g., Panduit Corp. v. Dennison Mfg. Co.*, 810 F.2d 1561, 1565-68 (Fed. Cir. 1987). In determining whether a claim is obvious, the Board considers (1) the level of ordinary skill in the pertinent art; (2) the scope and content of the prior art; (3) the differences between the claimed invention and the prior art; and (4) objective secondary considerations of non-obviousness, if any. *See, e.g., SIBIA Neurosciences, Inc. v. Cadus Pharm. Corp.*, 225 F.3d 1349, 1355 (Fed. Cir. 2000); *In re Gartside*, 203 F.3d 1305 (Fed. Cir. 2000). The person of ordinary skill in the art ("POSITA") is a hypothetical person who is presumed to know the relevant prior art. *See Custom Accessories, Inc. v. Jeffrey-Allan Indus., Inc.*, 807 F.2d 955, 962 (Fed. Cir. 1986).

Importantly, "[a] patent composed of several elements is not proved obvious merely by demonstrating that each element was, independently, known in the prior art." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 401 (2007). Rather, to establish *prima facie* obviousness, the cited references must be shown to disclose or suggest each claimed element *and* the Petition must establish that it would have been obvious to combine the teachings in the references together to arrive at the claimed invention. *See In re Lee*, 277 F.3d 1338, 1343 (Fed. Cir. 2002). The Petition must demonstrate

an apparent reason to modify a reference, or to combine the cited references to create the specific invention.

The rationale asserted by the Petition must follow from the nature of the problem to be solved, it must be "clear and particular, and it must be supported by actual evidence." *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1334 (Fed. Cir. 2002). The combined art must narrow the scope of the inventor's task to the point where it would have been obvious to try the particular invention claimed. If the particular invention lies hidden in a multitude of other options suggested by the art, the invention is not obvious. *See, e.g., Leo Pharm. Prods. Ltd. v. Rea*, 726 F.3d 1346, 1356 (Fed. Cir. 2013); *In re Kubin*, 561 F.3d 1351, 1359-60 (Fed. Cir. 2009).

C. Claim Construction – Order of Steps

A patent claim "requires an ordering of steps when the claim language, as a matter of logic or grammar, requires that the steps be performed in the order written, or the specification directly or implicitly requires' an order of steps." *Mformation Techs., Inc. v. Rsch. in Motion Ltd.*, 764 F.3d 1392, 1398-400 (Fed. Cir. 2014) (holding in part that "[f]urther, we note that other sub-steps in claim 1 inherently require an order-of-steps. As a matter of logic, a mailbox must be established before the contents of said mailbox can be transmitted"); *see also Function Media, LLC v. Google, Inc.*, 708 F.3d 1310, 1320 (Fed. Cir. 2013) (holding that a claim that recites

"processing" an "electronic advertisement" necessarily indicates that "the creation of the ad must happen before the processing begins") (emphasis added).

The Board followed the *Mformation* decision in a recent decision involving technology related to producing multiple, sexed embryos. *Trans Ova Genetics v. Xy, LLC*, No. IPR2018-00249, Paper 7 (PTAB June 25, 2018). In arriving at its decision, the Board relied on the principles of antecedent basis to find that the claim at issue required an ordered sequence of steps. Specifically, the Board held that:

We agree with Patent Owner that the **sorting limitation must be performed separately from and subsequent to the determining limitation**. In claim 1, the word "said" appears twice in the limitation "sorting according to said sex of said sperm cells." In each instance, the word "said" refers back to the prior limitation – i.e., the determining limitation. Thus, "said sex" and "said sperm cells," refer to the sex of the sperm cells mentioned in the previous step of "determining a sex of a sperm cell." Put another way, "sex" and "sperm cell" in the determining limitation serve as the **antecedent basis** for "said sex" and "said sperm cells" in the sorting limitation. Accordingly, we agree with the Patent Owner that "as a 'matter of logic or grammar,'" the "determining" step must occur before the "sorting" step, not after it or simultaneously with it.

Id. at 8 (internal citations omitted) (emphasis added); *see also Unified Patents Inc. v. Divx LLC*, No. IPR2019-01379, Paper 52 at 24 (PTAB Feb. 8, 2021) (holding that

"Petitioner's claim construction disregards Federal Circuit precedent regarding antecedent basis and renders superfluous the second portion of the claim limitation at issue").

IV. PETITIONER'S INVALIDITY GROUNDS FAIL TO ESTABLISH THAT ANY CHALLENGED CLAIM IS UNPATENTABLE

A. Petitioner's Alleged Invalidity Grounds

Petitioner alleges that all the Challenged Claims are obvious over the following single reference or combination of various references:

Ground 1: Claims 1-5, 9, 19-21, and 26 – Khandekar (Ex. 1004);

Ground 2: Claims 1-5, 9, 19-21, and 26 – Khandekar and Le (Ex. 1005);

Ground 3: Claims 9, 10 – Khandekar, Le, and Sidebottom (Ex. 1006); and

Ground 4: Claim 17 – Khandekar, Le, and Stone (Ex. 1007).

See Pet. at 9.

B. Person of Ordinary Skill in the Art

A POSITA at the time of the invention of the '138 Patent would have had a bachelor's degree in Electrical Engineering, Computer Science, or a similar discipline, with one to two years of experience in this or a related field. *See* Ex. 2001, ¶ 24. Petitioner relies on a similar description of a POSITA in its Petition. *See* Pet. at 12-13; *see also* Ex. 1003 ¶¶ 33-35. Any differences in the parties' view of a POSITA are not material to the arguments herein. *See* Ex. 2001, ¶ 25.

C. Claim Construction – Challenged Claims Require an Ordered Sequence of Steps

In its Institution Decision, the Board stated that "[t]he parties are encouraged to address, in the response and reply, whether claim 1 must be construed to require the performance of an ordered sequence of steps as argued by the Patent Owner." Institution Decision at 26 n.9 (Paper 9). In addition, the Board stated that "[a]s with claim 1, the parties are both encouraged to address, in the response and reply, whether claims 19 and 26 must be construed to require the performance of an ordered sequence of steps, as argued by Patent Owner." *Id.* at 28 n.12.

As stated above, a claim requires an ordering of steps when (1) the "**specification** directly or implicitly requires an order of steps" or (2) "**claim language**, as a matter of logic or grammar, requires that the steps be performed in the order written." *See Mformation Techs., Inc.*, 764 F.3d at 1398-400 (emphasis added). Here, **both** the specification and the claim language dictate that each of the three of the independent Challenged Claims require an ordered sequence of steps.

1. The Specification of the '138 Patent

The '138 Patent provides a detailed description of both the capture and deployment processes for the image instances of the virtual machine manager and the image instances of the software applications. Importantly, the '138 Patent also makes clear that the image instances of the virtual machine manager provide (create)

the virtual machines upon which the control node deploys the image instances of the software applications.

a) Two-Phase Capture Process

To create the image instances of the virtual machine manager and the software applications, the '138 Patent provides a description of a "two-phase capture process" to first capture image instances of the virtual machine manager and then the image instances of the software applications. *See* Ex. 1001 at 33:42-45. During his deposition, Petitioner's expert Dr. Prashant Shenoy, agreed that the '138 Patent disclosed a two-phase process for capturing these images. Ex. 2002 at 40:22-41:8 (testifying that "Yes. In this embodiment that we are discussing, it does refer to a two-phase capture process for capturing the virtual machine manager and the applications.").

(1) Virtual Machine Manager

During the first capture process, the '138 Patent makes clear that the image instances of the virtual machine manager are created **before** the installation of any software applications. Specifically, the control node installs a virtual machine manager onto a computer (*e.g.*, a node) with particular settings and then captures the image of this particular type of virtual machine manager. Ex. 1001 at 33:35-54. This capture process of the virtual machine manager is done prior to any installation and/or configuration of the software applications. *Id.* (explaining that "control

node 12 may first capture an image of virtual machine manager 402 installed on physical node 400 **prior to** installation and configuration of any guest operating system and applications 406"). Dr. Shenoy agreed with this assessment of the first capture process:

Q Okay. And it states that the first capture is of the image instance of the virtual machine manager; correct?

A So at line 46 [of column 33], it does talk about control node 12 would capture -- would first capture an **image of the virtual machine manager** installed on physical node 400. So in this particular example, it talks about **doing that first**.

Q And that capture is done prior to the installation and configuration of any guest operating system and applications; correct?

A So, again, that sentence goes on and says it's **prior to** the installation and configuration of any guest operating system and applications 406. So in this particular example, yes.

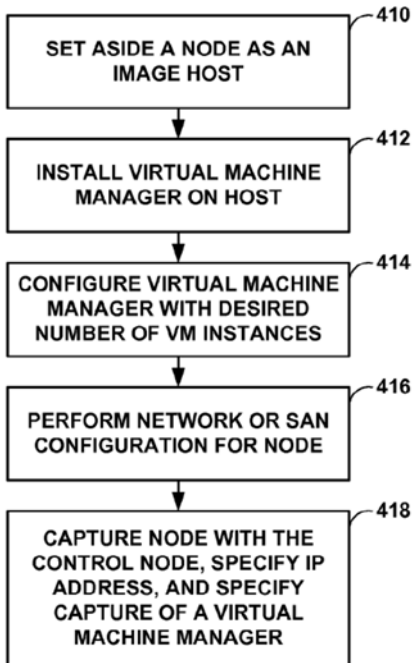
Q Okay. So to put this in laymen's terms, the virtual machine manager 402 is installed on a physical node 400, and the control node 12 captures an image instance of that particular virtual machine manager; correct?

A So, in this example, the control node is capturing an image of the virtual machine manager where there is no OS or applications installed.

Ex. 2002 at 41:9-42:9 (emphasis added).

The '138 Patent explains that this capture process of the image instance of a virtual machine manager allows the control node to create a "golden image" of that particular virtual machine manager. Ex. 1001 at 33:46-54. This golden image of a virtual machine manager can act as a master image to create other image instances of that particular type of virtual machine manager – a point that Dr. Shenoy agreed with in his deposition. Ex. 2002 at 42:18-43:1.

In Figure 25, replicated below, the '138 Patent sets forth a process for "creating and capturing a golden image for a particular virtual machine manager." Ex. 1001 at 34:48-50.



Id., Fig. 25. Dr. Shenoy described this process as follows:

Figure 25 is describing a specific embodiment where you first set aside or choose a node for imaging as an image host, that's step 410. Then you install a virtual machine manager on that host, which is 4 -- step 412. You can then configure the virtual machine manager with a desired number of VMs that you want to run on it. That's the next step. You can perform additional configuration of networking and storage area network, SAN, configuration. That's step 416. **And in the final step, you capture using the control node -- or you specify, rather, the capture of the virtual machine manager that is being installed on that host.**

Ex. 2002 at 47:3-22 (emphasis added); *see also* Ex. 1001 at 34:50-67.

At the conclusion of this first capture process, "[c]ontrol node 12 may store the captured image as a golden image in application matrix 350 for use in subsequent autonomic deployment of image instances." Ex. 1001 at 34:64-67. Dr. Shenoy acknowledged that storing the captured image enabled the control node to subsequently use these image instances of the virtual machine manager for autonomic deployment. Ex. 2002 at 48:1-49:2 (testifying that the "control node will store the captured image as a golden image and will use it subsequently for autonomic deployment").

Dr. Shenoy further agreed that image instances of different virtual machine managers can be useful because the software image repository can store image instances for virtual machine managers for each type of application node.

Q So in a hypothetical example, a person of ordinary skill may have two different physical configurations for two different application nodes and, therefore, want to have two different image instances of a virtual machine manager to be specific to each of those particular nodes; correct?

A In this hypothetical example where there are **two different application nodes with different configurations, you can capture images of the virtual machine manager for each of those application nodes.**

Ex. 2002 at 51:14-52:5 (emphasis added). This allows for significant flexibility to enable autonomic deployment of virtual machine managers (and as described below) and application software to all types of application nodes. *See* Ex. 2001, ¶¶ 53-54.

(2) Software Applications

After the control node captures an image instance of a particular virtual machine manager, the control node uses the virtual machine manager image instance to create a virtual machine (*i.e.*, an emulated computer platform) and then installs and configures an operating system and specific applications onto the virtual machine, and then captures an image instance of the software applications. *See* Ex. 1001 at 33:55-60. Similar to the first capture process, the '138 Patent states that:

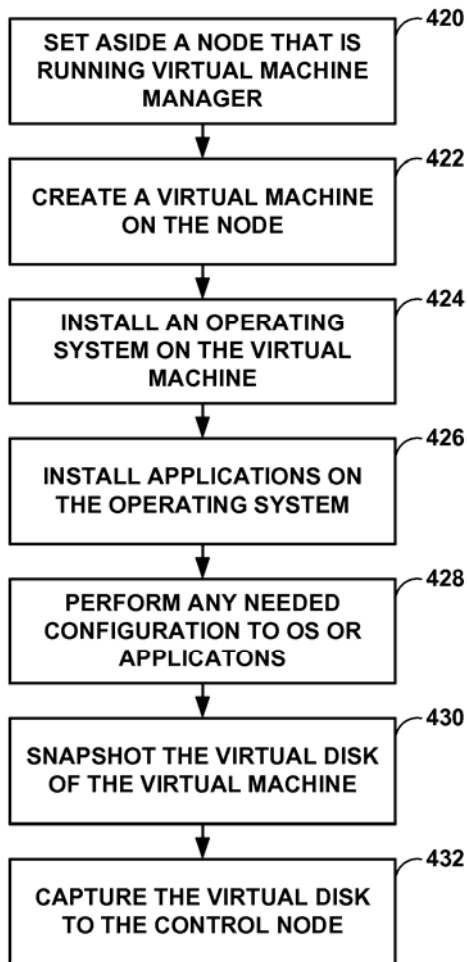
control node 12 may utilize the captured images as golden images for the particular operating systems, applications and specific configuration of the virtual machine. The golden image may then be used to produce image instances, *i.e.*, virtual disk images in this case, that are bootable on the virtual machines. The image instances may likewise be stored in application matrix 350 for autonomic deployment.

Id. at 33:61-67.

In Figure 26 of the '138 Patent, replicated below, the '138 Patent sets forth a process "for creating and capturing a virtual disk image file, *i.e.*, a golden image of

a guest operating system and applications for execution **on a virtual machine.**"

Ex. 1001 at 35:1-4 (emphasis added).



Id., FIG. 26. Dr. Shenoy described this process as follows:

A Yes. Figure 26 does describe an example procedure for creating and capturing a virtual disk image file, which is referred to as a golden image, of the **guest operating system and applications and for execution on the virtual machine.**

Q Okay. So I'm going to go, then, to figure 26. I have figure 26 on the screen. If you could, then, as you did with figure 27, kind of walk us through your understanding of what this flow chart describes.

A Okay. So this is an example procedure for **capturing image instance of software applications**. So it starts off by choosing or setting aside a node that is running a **virtual machine manager**. And it **creates a virtual machine** on that node. That's the second step. It then **installs an operating system and applications** and performs any configurations for the OS and applications as needed. Then it takes a snapshot of the virtual disk, of the virtual -- of the VM, and it captures the virtual disk image onto the control node.

Q And then, similar to figure 25, but with figure 26, once that image instance of the software applications is captured, then it is then stored for later autonomic deployment; correct?

A That is not shown in the flow chart, but the text that we had discussed earlier does talk about **storing this image [f]or autonomic deployment**.

Ex. 2002 at 56:15-57:15 (emphasis added); *see also* Ex. 1001 at 35:1-24.

As noted by Dr. Shenoy in the highlighted portions of this testimony above, this process of installing the software applications on a node to capture an image instance of the software applications requires the following three initial steps:

(1) select a node with a virtual machine running on the node, (2) use the image instance of the virtual machine manager to create a virtual machine, and (3) install the software applications on the virtual machine.

(3) Virtual Machines

Finally, as noted in the above passages, the '138 Patent states repeatedly that the image instances of the virtual machine manager create the virtual machines upon which the control node deploys the image instances of the software applications. For example, referring again to FIG. 26, the '138 Patent states:

Administrator 20 may then configure the virtual machine manager to **create a virtual machine** (e.g., virtual machine 404) on node 400 (422). **After creating the virtual machine**, administrator 20 installs the particular operating system on virtual machine 404 (424). Administrator 20 may then install one or more applications on the operating system (426).

Ex. 1001 at 35:6-12 (emphasis added); *see also id.* at 35:33-36 (stating that "[a]fter these physical nodes boot with the virtual machine manager image instance, each of the nodes **creates a set of virtual machines** as specified by the image instances.") (emphasis added).

In discussing these examples, Dr. Shenoy acknowledged that the image instances of the virtual machine manager create the virtual machines. With respect to the above quote related to Figure 26, Dr. Shenoy testified as follows:

Q And that's the same for this figure, right? If we look at, as an example, look at column 35, lines 6 through 10, in that section, it states that the virtual machine manager creates the virtual machine. For example, virtual machine 404 on node 400; correct?

A So just to be clear, the sentence that appears in column 35 is referring to figure 26. The figure on the screen is figure 24. So the figure 26 is a flow chart, and **it does say that virtual machine manager 402 can create a virtual machine.**

Ex. 2002 at 30:17-31:6 (emphasis added). Similarly, with respect to the above quote related to Figure 27, Dr. Shenoy testified:

Q Okay. And as used in the Sundar patent -- well, strike that. **The Sundar patent teaches that the virtual machine manager creates the virtual machine;** is that correct?

MR. LIANG: Objection.

A So -- so I'm going to refer to figure 27A, which is described in column 35, line -- starting line 33. And the sentence there says "After these physical nodes boot with the virtual machine manager image instance, each of the nodes creates virtual machine as specified by the imaged instances." **So in the context of figure 27A, it is describing a virtual machine manager instance, which allows the node to create virtual machines as specified.**

Id. at 26:6-22 (emphasis added). Finally, Dr. Shenoy was not aware of any other way that a virtual machine is created pursuant to the teaching of the '138 Patent other than being created by a virtual machine manager image instance.

Q Are you aware of anything in the Sundar patent that refers to a way that a virtual machine is created, other than being created by a virtual machine manager?

A **The two examples that we just discussed were ones where the virtual machine manager was creating the virtual machine.** Those were figures 26 and 27A. I have not read the patent, but **I cannot recall other disclosures that say otherwise from those two figures.**

Id. at 33:12-21 (emphasis added); *see also id.* at 44:19-46:16 (testifying that "[a]gain, that -- we looked at figure 26 and 27, **where the virtual machine manager was creating virtual machines.** We also talked about executing virtual machines").

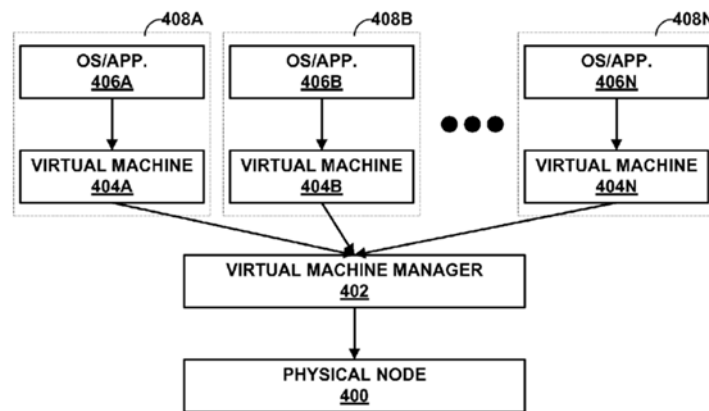
b) Three-Step Deployment Process

After describing how to capture and store separate image instances of the virtual machine manager and the software applications, the '138 Patent describes how the control node deploys these image instances onto application nodes. This deployment process is similar to the process of installing the software applications onto a node to capture an image instance of the applications as described above. *See supra* Section IV.C.1.a); Ex. 2001, ¶¶ 57-58.

(1) The Specification Requires Three-Step Deployment Process

The specification of the '138 Patent describes two different examples of a deployment process involving the image instances of the virtual machine manager and the software applications. Both examples require a three-step deployment process.

In the first example, the specification describes a deployment process to create the structure set forth in FIG. 24 below:



Ex. 1001, FIG. 24. In describing Figure 24, Dr. Shenoy noted that **"the virtual machine manager is a software program that manages virtual machines"** and the **"virtual machine manager is executing on that computer or physical node."**

Ex. 2002 at 16:19-18:2 (emphasis added). Dr. Shenoy further testified concerning Figure 24 that **"virtual machine in the context of the patent, and this example is software that creates an environment to emulate a computing platform."** *Id.* at 18:3-19:11 (emphasis added).

The '138 Patent goes on to describe the process used to create the structure set forth in Figure 24 as follows:

After generation of image instances for virtual machine manager 402 and virtual disk image files 408,² control node 12 may deploy the image instance of virtual machine manager 402 to one or more nodes in free pool 13. **Once control node 12 deploys the image instance of virtual machine manager 402 to a node, the node appears to be one or more new, unallocated nodes.** For instance, if the image instance of virtual machine manager 402 is configured to support five **virtual machines**, control node 12 represents the newly deployed image instance as though five new, unallocated nodes have been added to free pool 13. **Subsequently, control node 12 may deploy an image instance of virtual disk image files 408 to one of the new unallocated nodes.**

Ex. 1001 at 34:1-13 (emphasis added).

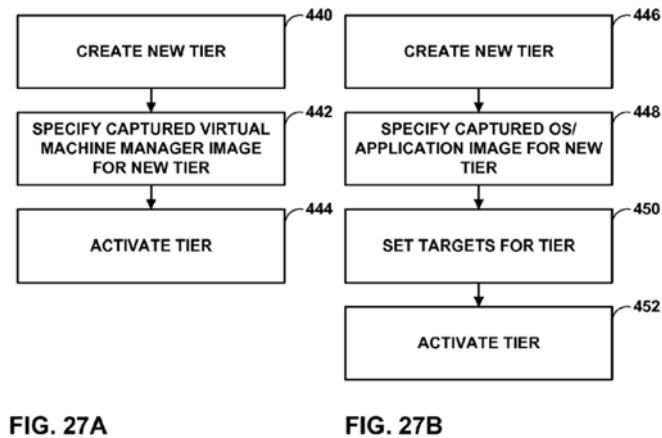
² As noted in dependent Claim 2, disk image files can be a type of an image instance of a software application. Ex. 1001 at 36:50-52.

The logic and grammar of this paragraph mandate this three-step deployment process.

- **Step 1** – To begin the deployment process, the specification uses the phrase "Once control node 12 deploys the image instance of virtual machine manager 402 to a node" (*i.e.*, this step must happen first). Ex. 1001 at 34:5-6.
- **Step 2** – The specification next states that the virtual machine manager supports the virtual machine(s) on the nodes (*i.e.*, the virtual machine manager provides the virtual machine on the node). *Id.* at 34:7-11. This of course cannot happen as a matter of logic until after the control node deploys the image instance of the virtual machine manager onto the node.
- **Step 3** – The specification states "Subsequently, control node 12 may deploy an image instance of virtual disk image files 408 to one of the new unallocated nodes" (*i.e.*, the image instances of the software applications are deployed onto the virtual machine after the deployment of the image instance of the virtual machine manager). *Id.* at 34:11-13 (emphasis added); *see also id.* at 32:67-33:2.

Ex. 2001, ¶¶ 67-68.

In the second example, the specification contains two flow charts in Figures 27A and 27B that describe the same type of deployment process.



Ex. 1001, FIG. 27.

In describing Figure 27A, the specification explains:

FIG. 27A is a flowchart illustrating an exemplary procedure for **autonomic deploying a virtual machine manager to a node**. Initially, administrator 20 may instruct control node 12 to create a new tier (440). Subsequently, administrator 20 specifies one or more **captured virtual machine manager image instances** for assignment to the slots of the new tier (442). Administrator 20 may also specify whether the image instances of the tier run in persistent or non-persistent mode. Administrator 20 may then activate the new tier (444). After these physical nodes boot with the virtual machine manager image instance, each of the nodes creates a set of virtual machines as specified by the

image instances. Control node 12 may then discover each of these virtual machines and place them in free pool 13 as if they were independent nodes. In this manner, each of the virtual machines appear to be physical nodes available for use as application nodes 14 when needed.

Id. at 35:25-47 (emphasis added); *see also* Ex. 2001, ¶¶ 62-63.

As highlighted above, this paragraph describes the first two steps of the deployment process: (1) the control node deploys the image instances of the virtual machine manager to the node and (2) the image instance of the virtual machine manager executes (or boots) on the node to create the virtual machine. *See* Ex. 2001, ¶¶ 62-63. Once the virtual machine manager image instance creates the virtual machine on the node, the control node places the virtual machine in the "free pool" to be used for the later deployment of the image instances of the software applications (*i.e.*, the third step in the deployment process). Ex. 1001 at 35:36-40; *see also* Ex. 2001, ¶¶ 58, 62.

In describing Figure 27B, the specification states that "FIG. 27B is a flowchart illustrating an exemplary procedure for **deploying an operating system and application image instance to a node that is executing a virtual machine manager.**" Ex. 1001 at 35:48-51 (emphasis added). The specification goes on to state that:

control node 12 autonomically selects a **virtual machine from free pool 13**, associates the virtual machine with a slot in the tier, identifies the virtual disk image file (image instance) associated with the particular slot, **copies the virtual disk image file to the local hard disk of the physical node on which the virtual node resides** or an associated SAN, and finally boots the virtual machine from the virtual disk image file.

Id. at 35:60-67 (emphasis added). In other words, after the first two steps that create the virtual machine on a node, the control node then selects the node (with the virtual machine) from the free pool and deploys an image instance of the software application onto the virtual machine – a separate and independent step from the prior two steps.

Therefore, as set forth in multiple examples in the specification, the autonomic deployment process disclosed in the '138 Patent requires a three-step deployment process. *See* Ex. 2001, ¶ 66.

(2) Dr. Shenoy Concedes the Three-Step Deployment Process

Dr. Shenoy conceded that the specification of the '138 Patent expressly requires this three-step deployment process as shown in the following excerpts from his deposition.

Deployment process described in Sundar at 34:1-13	Q So in this particular example that we've just discussed in column 34, lines 1 through 13 of the Sundar patent, it talks about deploying the image instances of the virtual
---	--

	<p>machine manager and the image instances of the software applications in two different steps; correct?</p> <p>A So in this particular example, it does talk about deploying the virtual machine manager node and <u>then</u> deploying image instances of virtual disk image files to an unallocated node.</p> <p>Ex. 2002 at 63:10-64:6 (emphasis added).</p>
Deployment process described in FIG. 27A	<p>Q Okay. So in this process, once the image instance of the virtual machine manager is deployed to a node in the free pool and the virtual machine manager is used to create a virtual machine, then that physical machine then becomes available for use as an application node 14; correct?</p> <p>MR. LIANG: Objection to form.</p> <p>A So in this case, <u>once</u> the virtual machine manager has been installed on the physical node and a virtual machine has been created, the virtual machine appears to be a physical node for use as application nodes.</p> <p><i>Id.</i> at 68:12-69:2 (emphasis added).</p>
Deployment process described in FIG. 27B	<p>Q Okay. And that first sentence talks about the flowchart that illustrates an exemplary procedure for deploying an operating system and application image instance to a node that is executing a virtual machine manager; correct?</p> <p>A So this is an example procedure for <u>deploying an OS and an application image to a node that's executing a virtual machine manager.</u></p> <p>Q So in this particular example, the node is running a virtual machine manager, and this flow chart is talking about a procedure for deploying the operating system and the application image instance on top of that node that's running the virtual machine manager; correct?</p>

	<p>A <u>So in this example, you are <u>deploying an OS and an application image to a node that is running a virtual machine manager.</u></u></p> <p>Ex. 2002 at 71:6-72:6 (emphasis added).</p> <p>Q So in these examples -- or this example that we've shown in Sundar, it appears that Sundar teaches that you store the image instances of the virtual machine manager and the image instance of the applications separately so they can be deployed in two different steps; correct?</p> <p>MR. LIANG: Objection to form.</p> <p>A <u>So certainly the virtual machine manager image and the image of the software applications are stored separately in this example on the image repository, and then they are deployed. In this example, the virtual machine manager image is deployed, and then the virtual disk image is deployed.</u></p> <p>Ex. 2002 at 73:2-74:3 (emphasis added).</p>
--	--

Therefore, in summary, Dr. Shenoy concedes the following deployment process described in the specification of the '138 Patent:

- **Step 1** – The image instance of the virtual machine manager deploys before the image instance of the software applications. Ex. 2002 at 63:10-64:6; 68:12-69:2; 73:2-74:3.
- **Step 2** – Once the image instance of the virtual machine manager is deployed on the node, it creates the virtual machine. *Id.* at 68:12-69:2; *see also id.* at 73:2-74:3 (testifying that the control node is "deploying

an OS and an application image to a node that's executing a virtual machine manager" (*i.e.*, the virtual machine manager creates a virtual machine when it is executing on a node)).³

- **Step 3** – The image instance of the software application (*e.g.*, virtual disk) deploys after the deployment of the virtual machine manager and after the creation of the virtual machine. The image instance of the software application deploys onto the virtual machine (*i.e.*, the unallocated node that is executing the virtual machine manager). *Id.* at 63:10-64:6; 73:2-74:3.

Patent Owner's expert, Dr. David Rosenblum, agrees with Dr. Shenoy's testimony concerning the three-step deployment process set forth in the specification of the '138 Patent. *See* Ex. 2001, ¶ 69.

2. The Language of the Challenged Claims

The Challenged Claims include independent Claims 1, 19, and 26. Claim 1 covers a distributed computing system. Claim 19 covers a method. Claim 26 covers

³ *See id.* at 16:19-18:2 (describing Figure 24 that depicts a virtual machine manager, which created several virtual machines, as "[s]o in figure 24 of -- the physical node is an application node on a physical computing device. The virtual machine manager is executing on that computer or physical node") (emphasis added).

a computer-readable medium. Each of these independent Challenged Claims requires the three-step deployment process (or include claim elements that must perform the three-step deployment process) described in the specification of the '138 Patent.

a) Claim 1

Claim 1 states as follows:

1. A distributed computing system comprising:

a software image repository comprising non-transitory, computer-readable media operable to store: (i) a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes, wherein **when executed on the applications nodes, the image instances of the virtual machine manager provide a plurality of virtual machines**, each of the plurality of virtual machine operable to provide an environment that emulates a computer platform, and (ii) a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines; and

a control node that comprises an automation infrastructure to provide autonomic deployment of the plurality of image instances of the

virtual machine manager on the application nodes by **causing the plurality of image instances of the virtual machine manager to be copied from the software image repository to the application nodes and to provide autonomic deployment of the plurality of image instances of the software applications on the virtual machines by causing the plurality of image instances of the software applications to be copied from the software image repository to the application nodes.**

Ex. 1001 at 36:26-49 (emphasis added). While Claim 1 is indisputably a system claim, the logic and grammar used in the claim language mandate that the claim elements must perform the three-step deployment process set forth above.

(1) Separate Storage of Image Instances

First, the claimed software repository stores at least two separate sets of image instances: (1) a plurality of image instances of a virtual machine manager and (2) a plurality of image instances of software applications. Dr. Shenoy conceded that Claim 1 required separate storage of these two sets of image instances:

Q So based on what we've discussed in the Sundar patent and these two particular elements of **claim 1**, does it appear to you, as a person of ordinary skill, that there are a plurality of image instances of the virtual machine manager and then a separate set of image instances of the

software applications that are both stored on the software image repository?

A So first of all, figures 25 and 26 are flow charts or processes of depicting methods. It says **claim 1** is talking about a system. Having said that, it does talk about storing a plurality of image instances of VMMs and a plurality of image instances of software applications.

Q And based on these claim elements, those two sets of image instances are separate images; correct?

MR. LIANG: Objection to form.

A In the context of what we've been discussing, **the image instances of the virtual machine manager are separate from the image instances of software applications**, at least in the example embodiment that we have been discussing.

Q Okay. And there's nothing that would indicate in claim 1 that it is something different than in the embodiments that we've been discussing that are set forth in the Sundar patent; correct?

A **So in this case, the image instances of VMMs are different from the image instances of software applications.**

Ex. 2002 at 78:16-80:2 (emphasis added).

Therefore, based on this testimony from Petitioner's expert, it is undisputed that Claim 1 requires that the software image repository stores the image instances of the virtual machine manager separately from the image instances of the software applications.

(2) Image Instances of Software Applications are Deployed "on the virtual machines"

Second, Dr. Shenoy also conceded that within the context of Claim 1 the image instances of the virtual machine manager must be capable of providing a virtual machine on the application node and the image instances of the software applications must be capable of being executable on the virtual machines.

Q So would you agree with me, Dr. Shenoy, that the image instances of the virtual machine manager must be capable of creating a virtual machine on the software -- I'm sorry -- on the application nodes when executed?

A So when executed, **the virtual machine manager will provide one or more virtual machines on that application node.**

Q Okay. And then the plurality of image instances of the software applications must also be able to be executed on the virtual machines that can be created by the image instances of the virtual machine manager; correct?

MR. LIANG: Objection to form.

A So the claim doesn't say that either. It simply says that the plurality of software applications that are executable on the plurality of virtual machines.

Q And I might have asked a bad question, so I apologize if I did. But those image instances of the software applications that you just mentioned must be capable of being **executed on the virtual machines**. And in the context of **claim 1**, those **virtual machines are created when the plurality of virtual machine managers are executed**; correct?

MR. LIANG: Objection to form.

A So that was a long question, and I will repeat what I said earlier. **That when executed on the application nodes, the virtual machine manager provides a plurality of virtual machines. And also, that the image instances of software applications are executable on the virtual machines.**

Ex. 2002 at 84:22-86:11 (emphasis added).

While Dr. Shenoy argued that, as used in Claim 1, "executable is not the same as executing" (*id.* at 80:14-81:16), Claim 1 requires that the control node "provide autonomic deployment of a plurality of image instances of the software applications **on the virtual machines.**" *Id.* at 36:45-47 (emphasis added). There are two significant issues with respect to the phrase "on the virtual machines" that are part

of Claim 1 – one with respect to the term "on" virtual machines and the other with respect to the term "the" virtual machines.

First, in considering this limitation, it is important to understand that Claim 1 has two primary elements: a storage element related to the software repository and a deployment element related to the control node. *See* Ex. 2001, ¶ 80. The term "plurality of image instances of a virtual machine manager" is first set forth in the storage element. In describing the further limitations of this term, Claim 1 states that "**when executed** on the application nodes, the image instances of the virtual machine manager **provide a plurality of virtual machines.**" Ex. 1001 at 36:31-33 (emphasis added). Because this phrase is first found in the storage element of Claim 1, the phrase "when executed" is appropriate because when the plurality of image instances of the virtual machine manager are stored in the software repository, these images are obviously not executing on the application nodes.

However, within the context of the deployment element, Claim 1 requires that the control node provide autonomic deployment of the plurality of image instances of the virtual machine manager onto the application nodes and "provide autonomic deployment of a plurality of image instances of the software applications **on the virtual machines.**" As described in detail above, the only way to create virtual machines disclosed in the '138 Patent is for a virtual machine manager image

instance to execute on an application node. Ex. 2002 at 44:19-46:16; *see also supra* Section IV.C.1.a)(3); Ex. 2001, ¶¶ 64, 82.

Therefore, based on both logic and the grammar of Claim 1, the control node must first deploy the image of the virtual machine manager to create the virtual machine so that the image instances of the software applications can then be deployed on the virtual machine. *See* Ex. 2001, ¶ 83. Moreover, as described above, this process is exactly the same as how the specification of the '138 Patent describes the deployment process, which Dr. Shenoy conceded in his deposition. *See supra* Section IV.C.1.b)(2); *see also* Ex. 2001, ¶ 84.

Finally, although the specification states that the virtual machine manager image instances "create" a virtual machine (*see* Ex. 1001 at 35:33-36; *see also supra* Section IV.C.1.a)(3)) and Claim 1 states that the virtual machine manager image instances "provide" the virtual machine, the '138 Patent uses these terms interchangeably. For example, the '138 Patent states:

The term virtual machine is used herein to refer to software that **creates** an environment that emulates a computer platform. For example, virtual machines 40 **provide** environments on which guest operating systems (OS) execute as through the operating systems were operating directly on a physical computing platform.

Ex. 1001 at 32:67-33:6 (emphasis added). When asked about whether the term "create" and "provide" are used interchangeably in the '138 Patent, Dr. Shenoy was unable to provide an opinion. *See* Ex. 2002 at 22:4-23:2.

However, as the logic and grammar of the above quote from the '138 Patent make clear, the terms "provide" and "create" are plainly used interchangeably with respect to a virtual machine manager image instance creating/providing a virtual machine. *See* Ex. 1001 at 32:67-33:6 (emphasis added). Moreover, as acknowledged by Dr. Shenoy, he was not aware of any other way disclosed in the '138 Patent to bring a virtual machine into existence other than to create it using a virtual machine manager image instance. Ex. 2002 at 33:12-21; *see supra* Section IV.C.1.a)(3). Therefore, the specific embodiments in the specification describing the virtual machine manager image instance "creating" a virtual machine on an application node are claimed as part of Claim 1 "wherein when executed on the applications nodes, the image instances of the virtual machine manager **provide** a plurality of virtual machines." Ex. 1001 at 36:31-33 (emphasis added); *see also* Ex. 2001, ¶ 88.

Second, Claim 1 requires that the control node must provide autonomic deployment of the plurality of image instances of the software applications on **"the"** virtual machines. The term "a plurality of virtual machines" is defined earlier in Claim 1 with respect to the phrase "the image instances of the virtual machine

manager provide a plurality of virtual machines." Ex. 1001 at 36:31-33. Therefore, based on the well-known principle of antecedent basis, both phrases refer to the same virtual machines. *See Process Control Corp. v. HydReclaim Corp.*, 190 F.3d 1350, 1356-57 (Fed. Cir. 1999) (holding that "[t]he presence of that identical language clearly indicates that 'a discharge rate' in clause [b] is the same as 'the discharge rate' in clause [d], both referring to the rate (in units of weight per unit of time) that material is discharged from the common hopper to the material processing machine. In addition, that conclusion avoids any **lack of antecedent basis** problem for the occurrence of 'the discharge rate' in clause [d]") (emphasis added).

Claim 1 requires that the control node must provide autonomic deployment of the plurality of image instances of the software applications on "the" virtual machines (*i.e.*, the same virtual machines that must be created by the plurality of image instances of the virtual machine manager). Therefore, the control node of the distributed computing system of Claim 1 must provide automation infrastructure to first deploy an image instance of a virtual machine manager onto an application node. After the control node deploys the image instances of the virtual machine manager, the virtual machine manager image instance provides a virtual machine on the node. Finally, the control node must provide automation infrastructure to then deploy the image instance of the software applications on the virtual machine created by the virtual machine manager image instance.

b) Claim 19

Claim 19 contains similar language to Claim 1 with respect to the claim elements that require the three-step deployment process discussed above with respect to Claim 1 and the specification. *See supra* Section IV.C.2.a). As a result, this three-step deployment process is required by Claim 19 for the same reasons. *See id.*

However, because Claim 19 is expressly a method claim, it is even more apparent that this three-step deployment process is required by Claim 19. Indeed, with respect to Claim 19, Dr. Shenoy conceded in his deposition that:

- the plurality of image instances of the virtual machine manager are stored separately from the plurality of image instances of the plurality of software applications (Ex. 2002 at 98:16-99:3);
- the plurality of image instances of the software applications must be able to be executed on a plurality of virtual machines (*id.* at 99:4-13);
- the first execution step of Claim 19 performs operations to achieve the autonomic deployment of the image instances of the virtual machine manager (*id.* at 102:3-10); and
- the second execution step of Claim 19 performs operations to autonomically deploy image instances of the software applications on

to the virtual machines (*id.* at 102:19-103:17).⁴

See Ex. 2002 at 93:13-104:4.

As a result of these concessions by Dr. Shenoy, there can be no reasonable dispute that Claim 19 requires the three-step deployment process as evidenced by both the specification and the logic and grammar of the claim language itself.

c) Claim 26

Because Claim 26 covers the computer-readable medium that could be used to implement the claimed method of Claim 19 (*see* Ex. 1001 at 38:61-39:15), Claim 26 requires the same three-step method deployment process of Claim 19 for the same reasons.

D. Claim Construction – Petitioner's Proposed Claim Construction Attempts to Read the Deployment Process Out of the Challenged Claims

Petitioner argues for a construction of the phrase "a plurality of image instances of a plurality of software applications that are executable on the plurality

⁴ Finally, as with Claim 1, Claim 19 requires that after the first execution step, the image instances of the virtual machine manager provide the virtual machines that the image instances of the software application are deployed onto in this claimed method. Ex. 1001 at 38:4-6 ("wherein the image instances of the virtual machine manager provide a plurality of virtual machines").

of virtual machines" that would read out the three-step deployment process out of the Challenged Claims. Pet. at 13. Specifically, Petitioner argues that a "POSITA would have understood this term is satisfied by, but not limited to, a plurality of virtual disk image files. Dependent claim 2 specifies that 'the image instances of the plurality of software applications comprise virtual disk image files.' A teaching sufficient to satisfy claim 2 must also be sufficient to satisfy claim 1." *Id.*

There is no question that "virtual disk image files" can be a type of "image instances of the plurality of software applications" – dependent Claim 2 makes that clear. *See* Ex. 1001 at 36:50-52. These specific claim terms are not in dispute and do not need any construction because the language of Claims 1 and 2 make this point clear.

However, this is not the entirety of Petitioner's proposed construction. Indeed, Petitioner wants the Board to disregard the portion of this limitation that states "that are executable on the plurality of virtual machines." Specifically, the Petitioner is asking the Board to make the clear substitution: "a plurality of image instances of a plurality of software applications **that are executable on the plurality of virtual machines**" for "virtual disk image files." *Id.* at 36:36-38 (emphasis added); *see also id.* at 36:50-52. Claim 2 states the "a plurality of image instances of a plurality of software applications" can comprise "virtual disk image files," but importantly Claim 2 does not address how (or where) the control node deploys the virtual disk

image files. *See* Ex. 2001, ¶ 105.

As discussed above, the phrase "that are executable on the plurality of virtual machines" (and the similar phrase "on the virtual machines" in the deployment element) play a significant role in dictating the three-step deployment process of the Challenged Claims. *See supra* Section IV.C.1.a)(2). The reason for this is simple – "on the plurality of virtual machines" dictates where the image instances of the software applications must be deployed (*i.e.*, "on" the virtual machines that are on the application nodes after the second step of the deployment process described above). *Id.*; *see also* Ex. 2001, ¶ 107.

Moreover, the phrase "that are executable on the plurality of virtual machines" refers to the "a plurality of virtual machines" that previously define "virtual machines." If adopted by the Board, Petitioner's disregard for the principles of antecedent basis in its proposed construction would render the phrase "that are executable on the plurality of virtual machines" superfluous – an attempted claim construction tactic roundly rejected by the Federal Circuit and the Board in prior decisions. *See, e.g., Mformation Techs., Inc.*, 764 F.3d at 1399 (holding that "[w]e are persuaded by BlackBerry's argument that the separate sub-step for establishing a connection would become 'superfluous' if we concluded that a connection did not have to be established (completed) before transmission. That is because, under such construction of the claim, establishing a connection is necessarily encompassed in

transmitting a command"); *Aristocrat Techs. Australia Pty Ltd. v. Int'l Game Tech.*, 709 F.3d 1348, 1356-57 (Fed. Cir. 2013) (rejecting appellants' proposed construction because it would render another limitation "superfluous"); *Unified Patents Inc. v. Divx LLC*, No. IPR2019-01379, Paper 52 at 24 (PTAB Feb. 8, 2021) (holding that "Petitioner's claim construction disregards Federal Circuit precedent regarding antecedent basis and renders superfluous the second portion of the claim limitation at issue"); *Trans Ova Genetics v. Xy, LLC*, No. IPR2018-00249, Paper 7 (PTAB June 25, 2018) (relying on antecedent basis to find that the cited claim elements require a specific ordered sequence of steps); *see also* Ex. 2001, ¶ 108.

Therefore, for these reasons, including the many reasons discussed above with respect to the ordered sequence of steps required by the Challenged Claims, the Board should reject the Petitioner's proposed construction and apply the plain and ordinary meaning to the term, "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines". *See* Ex. 2001, ¶ 109.

E. All Challenged Claims Are Valid Over Any Combination of Petitioner's Cited References of Khandekar, Le, Sidebottom, and Stone

1. Ground 1: Khandekar

With respect to Ground 1, Petitioner alleges that Khandekar renders independent Claims 1, 19, and 26, and dependent Claims 2-5, 9, 20, and 21, obvious

under 35 U.S.C. § 103. *See* Pet at 9. Each of these Challenged Claims are valid over Khandekar.

a) Overview of Khandekar

Khandekar discloses inventions related to a provisioning server that a user can interact with to obtain access to a virtual computer with user-requested features. Ex. 1004 at 9:26-27 (the "main feature of the invention is a VM provisioning server"). As Khandekar discloses:

A provisioning server automatically configures a virtual machine (VM) according to user specifications and then deploys the VM on a physical host. The user may either choose from a list of pre-configured, ready-to-deploy VMs, or he may select which hardware, operating system and application(s) he would like the VM to have. The provisioning server then configures the VM accordingly, if the desired configuration is available, or it applies heuristics to configure a VM that best matches the user's request if it isn't.

Id., Abs.; *see also id.* at 9:26-27; 9:65-10:1.

Importantly, the term "virtual machine" as used in the '138 Patent does not have the same meaning as the term "virtual machine (VM)" as used in Khandekar. As described above, the term "virtual machine" as used in the '138 Patent refers to "software that emulates a physical computing platform" on an application node. Ex. 1001 at 32:67-33:2; 36:34-36. A "virtual machine" in the '138 Patent emulates

a "bare-metal" computer without any operating system or applications installed. *See* Ex. 2001, ¶¶ 39, 131. A "virtual machine" in the '138 Patent is created by a virtual machine manager image instance on an application node. Ex. 1001 at 35:6-11 (emphasis added); *see also supra* Section IV.C.1.a)(3). In contrast, in Khandekar, "a 'model virtual machine' is a virtual machine (VM) that contains a certain configuration of guest operating system 320, service pack, and applications 360, where the applications have been tested and are known to work correctly." Ex. 1004 at 8:51-56.

While Khandekar discloses various types of VMs (pre-built, custom-built, and instantly deployable) (*see* Pet. at 41), none of these VMs are "software that creates an environment that emulates a computer platform" on a node. Ex. 1001 at 35:6-10. Rather, and as acknowledged by Dr. Shenoy, each type of VM is created in the provisioning server with an operating system and (in most cases) software applications. Ex. 2002 at 108:17-109:13. In other words, the term "VM" in Khandekar obviates any distinction between "software applications" and "virtual machine" in the '138 Patent by combining these separate elements into a single concept that is neither wholly one nor wholly the other. *See* Ex. 1001, FIG. 26 (showing the installation of the operating system (step 424) and applications (step 426) in the process of capturing an image instance of software

applications/virtual disk image); *see also supra* Section IV.C.1.a)(2); Ex. 2001, ¶¶ 114, 132.

Finally, Khandekar discloses a "'virtual machine monitor' (VMM)," which Khandekar describes as "usually a thin piece of software that runs directly on top of the host and virtualizes all, or at least some of, the resources of the machine, or at least of some machine." Ex 1004 at 7:4-9. However, Khandekar provides little disclosure of how the VMM is deployed. Indeed, the totality of this disclosure occurs in the following sections of Khandekar.

In the following description of the invention, merely for the sake of simplicity, only one VM/VMM pair is discussed. The discussion applies equally, however, to all such VM/VMM pairs that may be included in any given implementation of the invention. Moreover, **it will be assumed below that when a VM is installed on a host, then a corresponding VMM is either already installed on the host, or is included and installed together with the VM.**

Id. at 8:36-43 (emphasis added).

As is mentioned above, a VMM is typically included for each VM in order to act as its software interface with the underlying host system. **In the preferred embodiment of the invention, a VMM is already pre-installed on each host 1000, 1001 so as to ensure compatibility and proper configuration.**

...

It would also be possible, however, for a corresponding VMM to be included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known. An advantage of pre-installing a VMM, without a VM, on a plurality of hosts, such as hosts 1000, 1001, 1003, is that it will then be possible to deploy a given VM on any arbitrary one (or more) of these hosts. Moreover, as long as the VMM on each host exports a known hardware interface to the VM deployed on it, then VM deployment will be substantially independent of the actual physical hardware configuration.

Id. at 13:17-35 (emphasis added).

These differences between the meaning and use of the terms VM and VMM in Khandekar as opposed to "virtual machine," "virtual machine manager," and "software applications" in the '138 Patent are important to properly understand the significant shortcomings in Petitioner's obviousness arguments as described below.

See Ex. 2001, ¶ 116.

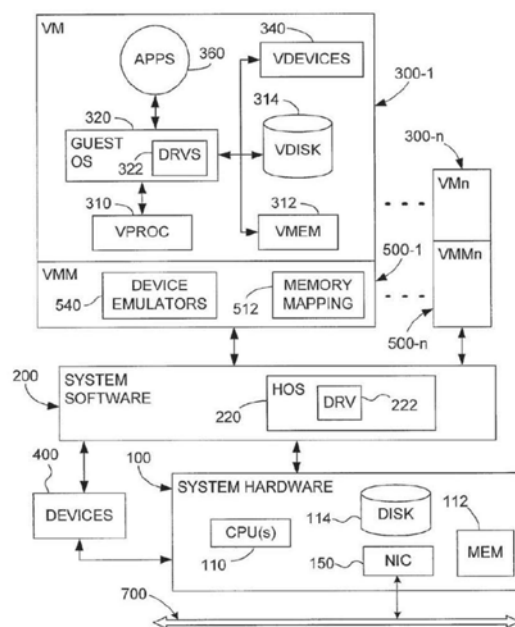
b) Khandekar Does Not Disclose the Three Step Deployment Process Required by the Challenged Claims

Khandekar does not disclose the three-step deployment process required by the Challenged Claims. *See supra* Section C.2.a). Rather, Khandekar teaches that

the VMs are assembled on the provisioning server (Ex. 2002 at 108:17-109:13) and then deployed onto the host is a single step. Indeed, Khandekar states that the deployment process is a single step: "'Deployment' is the step of taking a staged VM, copying it to a host machine and letting it configure with a new identity." Ex. 1004 at 8:64-67 (emphasis added).

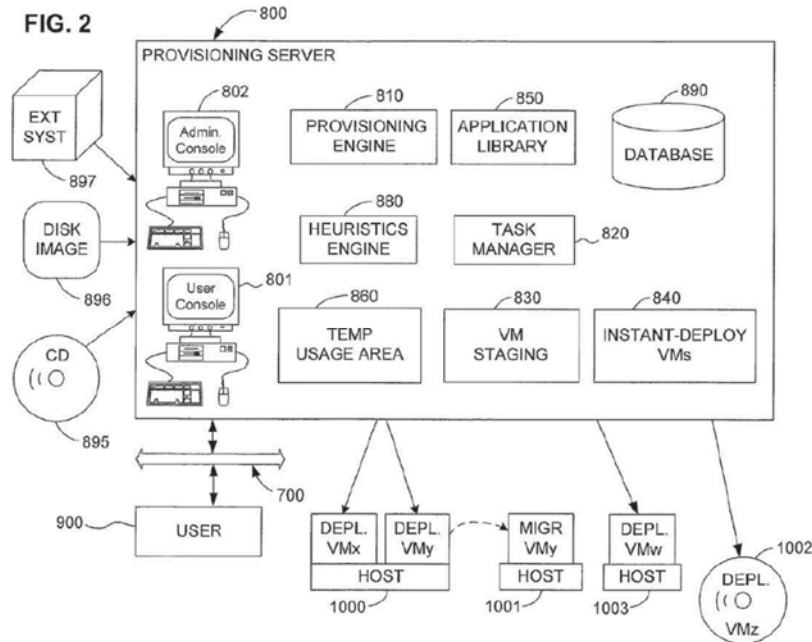
The shortcomings of Khandekar to disclose the specific limitations of Challenged Claims are clear with respect to Petitioner's invalidity allegations for what it denotes as elements [1d] and [1e]. *See* Pet. at 37-47. With respect to element [1d] that requires, *inter alia*, "when executed on the applications nodes, the image instances of the virtual machine manager provide a plurality of virtual machines," the Petitioner relies on Figure 1 of Khandekar shown below:

FIG. 1
(Prior Art)



See Pet. at 37-38. However, with respect to element [1e] that requires a "plurality

of image instances of a plurality of software applications that are executable on the plurality of virtual machines," the Petitioner relies on Khandekar's disclosure related to pre-built, custom-built, and instantly deployable VMs – examples of which are shown in Figure 2:



Importantly, Figure 1 of Khandekar is not the same as the invention disclosed in Figure 2. In fact, Figure 1 is labeled "Prior Art" and Khandekar's discussion related to this figure provides no disclosure of how the VMMs (500-1 and 500-n) are deployed. *See* Ex. 2001, ¶ 120. Therefore, Petitioner does not explain how Khandekar's disclosure of Figure 1 satisfies how VMMs (500-1 and 500-n) deploy "a plurality of virtual machines" on the application nodes. *See id.* Importantly, the significance of element [1d] of Claim 1 relates to how the virtual machine is created – none of which is disclosed in Khandekar's discussion of its "Prior Art" Figure 1.

See id.

Petitioner then argues that Khandekar's disclosure related to pre-built, custom-built, and instantly deployable VMs satisfies element [1e], which requires a "plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines." Pet. at 41-42 (emphasis added). However, any such argument is controverted by Khandekar itself as well as Petitioner's own concessions regarding the proper construction of claim element [1e]. *See* Ex. 2001, ¶ 134.

First, with regard to the "instant deploy" VMs, the Khandekar specification explains that this disclosure does not describe deployment at all; but rather the reallocation of already-running VMs. *See* Ex. 2001, ¶ 135. Second, with regard to the "pre-built" VMs, Khandekar discloses that, "[i]n the preferred embodiment of the invention, pre-built VMs will be stored with their operating systems and applications installed" Ex. 1004 at 15:43-45. As previously addressed in Section IV.D. *supra*, this directly contradicts the '138 Patent claims which require that the software application image instances be deployed on the virtual machines as the final step of the three-stage deployment process. *See* Ex. 2001, ¶ 135.

Third, with regard to the "custom built" VMs, Khandekar discloses that "custom built VMs will be stored with just their operating systems installed." Ex. 1004 at 15:45-46. Khandekar also teaches, however, the use of an "executable

installation file," *i.e.*, the file used to install an application for the first time. *Id.* at 18:61-19:2. This installation file, however, is not equivalent to the "image instances of software applications" required by the '138 Patent. *See* Ex. 2001 at 137. Netflix admits that an "image instance" as used in the '138 Patent refers to "a snapshot, or copy, of **installed and configured software**." *See* Ex. 2003 at 2 (emphasis added). An executable installation file is completely different, and a POSITA would not understand them to be equivalent. *See* Ex. 2001, ¶ 137. Thus, none of Khandekar's disclosures related to pre-built, custom-built, and instantly deployable VMs satisfy element [1e]'s requirement for a "plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines."

Finally in addition to the fact that Khandekar fails to disclose claim element [1e] as described above, Petitioner never explains how Khandekar's disclosure of the VMs deployed from the provisioning server – which is disclosed as a single deployment step (Ex. 1004 at 8:64-67) – can be deployed on "the virtual machines" supposedly disclosed in the "Prior Art" portion of Khandekar's disclosure. The reason for this is simple: Petitioner ignored the antecedent basis established in the Challenged Claims that requires the "virtual machines" established in claim element [1d] to be the same "virtual machines" referenced in element [1e] (as well as element [1g], which requires the image instances of the software applications be deployed "on the virtual machines"). *See* Ex. 2001, ¶¶ 122-23.

Petitioner's failure to consider antecedent basis is not simply attorney argument. It is a fact expressly conceded by Petitioner's expert.

Q Are you familiar with the legal term "**antecedent basis**"?

A **No, I am not.**

Q Okay. **But you would agree with me that the first time that the term "virtual machines" appears in claim 1 of the Sundar patent is in line 33 where it says "a plurality of virtual machines"; correct?**

A Yeah. If you ignore the words "virtual machine" in the context of the virtual machine manager that's also in the claim, the first time the words appear is in line 33 where it talks about a plurality of virtual machines.

Q Okay. **So when it says in line 40 -- the very end of line 46 and the beginning of line 47, where it says "on the virtual machines," doesn't "the virtual machines" there refer to the virtual machines that were first described in line 33 of claim 1 of the Sundar patent?**

A **So I haven't given an opinion on that issue in my declaration.** You know, I would have to think about whether that term "virtual machine" in that sentence is referring to the plurality of virtual machines, because the previous term talks about plurality of virtual machines on many different nodes.

Q As you sit here today, are you aware of any reason why a person of ordinary skill reading claim 1 of the Sundar patent wouldn't consider the phrase "the virtual machines" in line 47 of column 36 to mean the same thing as "a plurality of virtual machines" in line 33 of column 36 of the Sundar patent?

A So, again, **I did not opine on that particular issue.** The first part of the claim talks about a software image repository. It talks about images in that repository that, when executed, are executable on a plurality of virtual machines. The second part of the claim talks about a control node that is performing deployment. So **I do not have an opinion on that issue at this point.**

Q Okay. And as you sit here today, you have no anticipation of offering opinion on that issue; correct?

A Again, as far as today's deposition is concerned, I do not have an opinion on that matter. If the Netflix lawyers ask me to think about that issue carefully and offer additional opinions, I may take some time to do that and might have an opinion later on.

Q Okay. And so you -- and just to be clear, **you did not offer an opinion on that issue in your declaration,** which has been marked as Exhibit 1 in this deposition; correct?

MR. LIANG: Objection to the form.

A So as far as the issue of the words "virtual machines" that appear on line 47, whether they refer to the same virtual machines that are mentioned in line 33, I have not offered an opinion in my declaration.

Ex. 2002 at 90:17-93:12 (emphasis added); *see also id.* at 100:2-101:18; 103:18-104:4 (also conceding that he had no opinion regarding antecedent basis of the term "virtual machine" with respect to Claim 19).⁵

By refusing to acknowledge that "a plurality of virtual machines" in claim element [1d] are "the virtual machines" in elements [1e] and [1g], Dr. Shenoy attempts to avoid acknowledging the obvious fact that the Challenged Claims require the three-step deployment process – a process wholly absent from the single step deployment process disclosed in Khandekar. This approach allows both Dr. Shenoy and the Petitioner to mix and match different prior art disclosures to attempt to satisfy the claim elements while disregarding the required deployment sequence of the Challenged Claims. *See* Ex. 2001, ¶ 123. The Board should reject this attempt

⁵ Notably, in testifying about Khandekar, Dr. Shenoy ably employed the concept of antecedent basis to put a clause of Khandekar in context with respect to an earlier disclosure. *See id.* at 124:12-125:3 (testifying that "if I just read the whole thing in context, that phrase 'installation' is referring to some other text before it").

and find that Khandekar utterly fails to disclose the three-step deployment process required by the Challenged Claims.

c) Khandekar Fails to Disclose a Number the Elements of the Challenged Claims

Because of the substantial differences between the inventions of the '138 Patent and the disclosures of Khandekar, it is not surprising that Khandekar fails to satisfy many of the elements of the Challenged Claims.

(1) Khandekar fails to disclose a software image repository to store "a plurality of image instances of a virtual machine manager ... wherein when executed on the applications nodes, the image instances of the virtual machine manager provide a plurality of virtual machines ... to provide an environment that emulates a computer platform"

First and foremost, Khandekar does not even suggest a "software image repository" that stores "a plurality of image instances of a virtual machine manager" as set forth in Claim 1. *See* Ex. 1001 at 36:27-29. Rather, Khandekar states that its preferred method is to have the VMM "pre-installed on each host." *See* Ex. 1004 at 13:17-35. Petitioner did not rely on this preferred method of deploying the VMM to satisfy this element of Claim 1 because, in this embodiment, the VMM is preinstalled on the host, not deployed from a plurality of VMM image instances stored in a software image repository.

Instead, Petitioner relies on the single sentence in Khandekar that, "[i]t would also be possible, however, for a corresponding VMM to be included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known" (*see id.* at 13:25-28), to claim that "a POSITA would have found it obvious to likewise store a plurality of VMM images in the provisioning server so they may be deployed together to hosts, as taught by Khandekar." *See* Pet. at 27.

Petitioner supports its position that it would have been obvious to store image instances of virtual machine managers in the software repository with a single paragraph from Dr. Shenoy's declaration. *See* Pet. at 27; Ex. 1003, ¶ 83 (quoting Ex. 1004 at 8:36-44, 13:17-35) (quotations omitted). This proposition espoused by Dr. Shenoy, that "A POSITA would have found it obvious to follow these teachings from Khandekar, with VMM images being stored in the provisioning server and provided to hosts together with VM images," finds no support whatsoever in the actual disclosure of Khandekar. *See* Ex. 2001, ¶¶ 140-41. The isolated, single sentence of disclosure in Khandekar, that "[i]t would also be possible, however, for a corresponding VMM to be included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known," Ex. 1004 at 13:25-28, provides no teaching or suggestion about how a VMM might be stored, let alone teaching "VMM images being stored in the provisioning server." Ex. 1003,

¶ 83. This analysis is unfounded speculation by Petitioner's expert, rather than any teaching from Khandekar. *See* Ex. 2001, ¶ 141. Indeed, given Khandekar's teaching that VMMs and VMs are paired (and in fact "form a single cooperating system,") a POSITA would find no reason to store a VMM separately from a VM, except by using hindsight in light of the disclosure of the '138 Patent. *Id.*, ¶¶ 140-41 (quoting Ex. 1004 at 7:40-42).

As Dr. Rosenblum notes, no reference cited by Petitioner, including the references cited as part of the grounds for review (Khandekar, Le, Sidebottom, and Stone) and other references mentioned in the Petition (such as Powell (Ex. 1014) and French (Ex. 1016)), even remotely suggests storing an image of virtual machine manager in a software repository, let alone storing a plurality of such images. *See* Ex. 2001, ¶ 130. In particular, Petitioner's reliance on French and Powell is misplaced; not only does Petitioner fail to identify those references in any Ground for review, but the portions of those references cited by Petitioner provide no support for the stated proposition. *See id.*; *cf.* Pet. at 20.

Finally, even if the Board were to find that it would have been obvious to store a plurality of image instances of a virtual machine manager in a software repository – a position that finds no support in the prior art cited by the Petitioner – there is no evidence that these image instances of the virtual machine manager would be capable of being executed on an application node to create a virtual machine (*i.e.*, an

emulation of a computer platform). The reason for this is that nothing in Khandekar discloses a virtual machine manager to create an emulated computer platform that serves as a platform for image instances of software applications to be deployed on. *See* Ex. 2001, ¶ 131. Instead, the VMs disclosed in Khandekar already include an operating system and/or software applications and, therefore, cannot constitute the "virtual machine" created by the execution of the virtual machine manager as required by Claim 1. *See id.*

(2) Khandekar fails to disclose "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines"

As explained above, there is a significant difference between the VMs disclosed in Khandekar and the virtual machines disclosed in Sundar. *See supra* Section IV.E.1.a). The VMs disclosed in Khandekar already contain an operating system and/or software application at the time they are deployed by the provisioning server. *Id.* Therefore, Petitioner relies on the VMs disclosed in Khandekar to satisfy both the "image instances of the plurality of software applications" and "virtual machine" portions of this claim element. *See* Pet. at 41-47 (citing exclusively to various disclosures related to the pre-built, custom-built, and instantly deployable VMs disclosed in Khandekar for both portions of this claim element).

As a result, regardless of whether the Board finds that the Challenged Claims require the three-step deployment process described above, Khandekar does not

disclose this element of Claim 1. This claim element requires both "virtual machines" created by the execution of a virtual machine manager and an "image instances of software applications" that deploy "on the virtual machines" – the VMs disclosed in Khandekar cannot be both. *See* Ex. 2001, ¶¶ 131-33.

(3) Khandekar fails to disclose a control node "to provide autonomic deployment of the plurality of image instances of the software applications on the virtual machines"

Finally, Khandekar does not disclose a control node that can provide "autonomic deployment of the plurality of image instances of the software applications on the virtual machines" as required by Claim 1. Ex. 1001 at 36:44-46. Once again, this claim element requires both "virtual machines" created by the execution of a virtual machine manager and "image instances of software applications" that deploy "on the virtual machines" – the VMs disclosed in Khandekar cannot be both. *See* Ex. 2001, ¶¶ 131-32.

d) Advantages of the Challenged Claims over Khandekar

It is important to note that the image capture and deployment techniques recited by the Challenged Claims provide significant advantages over the prior art, including Khandekar. Essentially, the methodology of the '138 Patent provides for separate remote deployment of image instances of a virtual machine manager and image instances of software applications.

Under the system of the '138 Patent, an image instance of a virtual machine manager can be provided to correspond to each known physical or virtual configuration of the application nodes. *See* Ex. 1001, FIG. 25; *see also id.* at 34:48-67 (describing a process for capturing a "golden image" of a virtual machine manager for a particular node configuration). Dr. Shenoy agreed with this assessment. *See* Ex. 2002 at 50:9-52:5 (acknowledging that a POSITA may want to have specific images of a virtual machine manager for each type of configuration of a physical node).

Moreover, an image instance of a software application can be provided to correspond to each type of virtual environment needed (*e.g.*, each image instance of a software application can have different operating systems, applications, etc.). *See id.* at 34:30-47 (describing different operating systems); *see also id.* at FIG. 26 and 35:1-24 (describing a process for capturing "a golden image" of an operating system and applications). Then, the appropriate image instance of a virtual machine manager for the application node can be selected as well as a separate image instance of software applications. *See id.*, FIGS. 27A-27B; *see also id.* at 35:25-67 (describing a process of selecting and deploying image instances of the virtual machine manager and software application). The selected image instance of the virtual machine manager can be copied to the application node and executed to

provide the virtual machine, and then the selected image instance of the software applications can be copied onto the virtual machine. *See id.*

This technology provides significant scalability enhancements over any of the techniques taught by Khandekar. For example, consider the alternative embodiment of Khandekar in which the VMM is included with the VM. In an environment with, for example, 100 different physical hardware configurations of host nodes and 100 different software configurations of VMs, a system in which a VMM were included with a VM combined would require storing 10,000 images to cover every possible configuration (*i.e.*, 100 image instances of each of the VMM configurations multiplied by the 100 image instances of each of the VM configurations). *See* Ex. 2001, ¶¶ 55, 146-47.

By contrast, the systems claimed in the '138 Patent, by abstracting the image instances of the virtual machine manager from the image instances of the software applications would require storage of only 200 images. *See id.*, ¶ 55. This is because the image instance of the virtual machine manager can be selected independent of, and then paired with, the image instance of the software application. This efficiency has significant consequences with regard to maintenance and updating of images. *See id.*, ¶ 147. These advancements are not possible with any of the embodiments described by Khandekar. *See id.*, ¶¶ 146-47. In fact, as noted above, Khandekar acknowledges that it is better to pre-install a VMM on each host (which is highly

inefficient) than attempt to deploy a VMM image instance with a software application image instance, because, according to Khandekar, the latter technique cannot accommodate differing configurations of the underlying nodes. *See id.*

2. Ground 2: Khandekar and Le

Petitioner further alleges that the combination of Khandekar and Le renders independent Claims 1, 19, and 26, and dependent Claims 2-5, 9, 20, and 21, obvious under 35 U.S.C. § 103. *See Pet.* at 9. As noted above, Khandekar fails to disclose or suggest each limitation of any independent claim of the '138 Patent. Le fails to remedy these deficiencies of Khandekar, and the Challenged Claims are equally valid over the combination of Khandekar and Le. *See Ex.* 2001, ¶¶ 148-152.

With regard to the installation of a VMM image instance, Le teaches only that, "[t]o power on and run the cloned virtual machine, all that the host computer needs is to have the appropriate VMM software installed." *Ex.* 1005 at 14:28-30; *see also Ex.* 2001, ¶ 149. Thus, Le, like Khandekar, presumes that a VMM has already been installed when discussing deployment, and provides no enabling disclosure of how the pre-existing installation of a VMM might have occurred. *See Ex.* 2001, ¶ 150. Le also appears to teach that the "cloned virtual machine," which would correspond to the software application image instance, is copied to the host before the VMM image instance is executed. *See id.* Certainly, the Petitioner fails to identify, and a

careful review of Le fails to reveal, any teaching or suggestion of the three-step deployment process required by the independent Challenged Claims. *See id.*, ¶ 151.

Accordingly, Le fails to supplement Khandekar in any material way with regard to independent Claims 1, 19, and 26, and those claims are valid over the combination of Khandekar and Le. *See id.*, ¶ 152. The remaining claims challenged by Ground 2 are valid at least by virtue of their dependence from valid base claims. Accordingly, the Board should reject Ground 2 for the same reasons it should reject Ground 1.

3. Ground 3: Khandekar, Le, and Sidebottom

Petitioner further allege that Claims 9 and 10 are obvious under 35 U.S.C. § 103, over the combination of Khandekar, Le, and Sidebottom. *See Pet.* at 9. Petitioner does not allege that Sidebottom provides any material disclosure with regard to Claims 1, 19, and 26. *See Ex. 2001*, ¶ 153-54. Those claims, therefore, are valid over the combination of Khandekar, Le, and Sidebottom for the reasons discussed *supra* Sections IV.E.1 and IV.E.2 with regard to Grounds 1 and 2. *See id.* Claims 9 and 10 are valid over the same combination at least by virtue of their dependence from Claim 1.

4. Ground 4: Khandekar, Le, and Stone

Petitioner further allege that Claim 17 is obvious under 35 U.S.C. § 103, over the combination of Khandekar, Le, and Stone. *See Pet.* at 9. Petitioner does not

allege that Stone provides any material disclosure with regard to Claims 1, 19, and 26. *See* Ex. 2001, ¶ 153-54. Those claims, therefore, are valid over the combination of Khandekar, Le, and Stone for the reasons discussed *supra* Sections IV.E.1 and IV.E.2. with regard to Grounds 1 and 2. Claim 17 is valid over the same combination at least by virtue of its dependence from Claim 1.

V. CONCLUSION

For the reasons set forth above, Avago respectfully submits that the Board should deny Petitioner's request for *inter partes* review of the Challenged Claims of the '138 Patent as Petitioner has failed to demonstrate that any Challenged Claim is invalid by a preponderance of the evidence as required by 35 U.S.C. § 316(e).

VI. CERTIFICATION PURSUANT TO 37 C.F.R. § 42.24(d)

Pursuant to 37 C.F.R. § 42.24(d), Patent Owner certifies that based on the "word count" feature of Microsoft Word, this Patent Owner's Response includes 13,992 words.

Respectfully submitted,

Dated: June 22, 2021

By: /s/Daniel S. Young/

Daniel S. Young, Reg. No. 48,277
Chad E. King, Reg. No. 44,187
ADSERO IP LLC D/B/A/
SWANSON & BRATSCHELL LLC
8210 Southpark Terrace
Littleton, CO 80120
(303) 268-0066 (telephone)
(303) 268-0065 (facsimile)

*Counsel for Patent Owner Avago Technologies
International Sales Pte. Limited ("Avago")*

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6, the undersigned hereby certifies that a copy of the foregoing **PATENT OWNER'S RESPONSE** was filed through the Patent Trial and Appeal Board End to End (PTAB E2E) electronic filing system on June 22, 2021 with a confirmation copy served via electronic mail, on the following counsel of record for Petitioner:

Harper Batts
SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
email: HBatts@sheppardmullin.com

Jeffrey Liang
SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
email: JLiang@sheppardmullin.com

Chris Ponder
SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
email: CPonder@sheppardmullin.com

SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
email: Legal-Netflix-Broadcom-IPRs@sheppardmullin.com

Dated: June 22, 2021

By: /s/Daniel S. Young/
Daniel S. Young
Registration No. 48,277