Trials@uspto.gov
571-272-7822

Paper 34
Entered: September 6, 2022

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.

---

IPR2021-00542
Patent 8,572,138 B2

---

Before KRISTEN L. DROESCH, KERRY BEGLEY, and NATHAN A. ENGELS, *Administrative Patent Judges*.

DROESCH, *Administrative Patent Judge*.

# JUDGMENT
Final Written Decision
Determining No Challenged Claims Unpatentable
*35 U.S.C. § 318(a)*

# I. INTRODUCTION

We have authority to hear this *inter partes* review under 35 U.S.C. § 6, and this Final Written Decision is issued pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons that follow, we determine that Petitioner fails to establish by a preponderance of the evidence that claims 11, 12, 14, 22, and 24 ("challenged claims") of U.S. Patent No. 8,572,138 B2 (Ex. 1001, "'138 Patent") are unpatentable.

## *A. Procedural History*

Netflix, Inc., ("Petitioner") filed a Petition requesting an *inter partes* review of the challenged claims of the '138 Patent. Paper 3 ("Pet"). Avago Technologies International Sales Pte. Limited ("Patent Owner") filed a Preliminary Response. Paper 7. Pursuant to 35 U.S.C. § 314, we instituted trial on September 7, 2021. Paper 11 ("Dec.").

After institution of trial, Patent Owner filed a Response (Paper 18, "PO Resp."), to which Petitioner filed a Reply (Paper 23, "Pet. Reply"), to which Patent Owner filed a Sur-reply (Paper 27, "PO Sur-reply").

Petitioner relies on a Declaration of Prashant Shenoy, Ph.D., (Ex. 1003) to support its Petition. Patent Owner relies on a Declaration of David Rosenblum, Ph.D., (Ex. 2001) to support its Patent Owner Response. Petitioner relies on a second Declaration of Prashant Shenoy, Ph.D., (Ex. 1029) to support its Reply.

Dr. Shenoy and Dr. Rosenblum were cross-examined during trial, and transcripts of Dr. Shenoy's deposition (Ex. 2008) and Dr. Rosenblum's deposition (Ex. 1031) are included in the record.

Oral argument was held on June 22, 2022. A transcript of the oral argument is included in the record. Paper 33 ("Tr.").

*B. Related Matters*

The parties indicate the '138 Patent is the subject of litigation in *Broadcom Corp. v. Netflix, Inc.*, Case No. 3:20-cv-04677-JD (N.D. Cal.) ("related district court proceeding"), which was filed as *Broadcom Corp. v. Netflix, Inc.*, Case No. 8:20-cv-00529-JVS-ADS (C.D. Cal.) and later transferred to the U.S. District Court for the Northern District of California. *See* Pet. 76; Paper 6, 1.

Claims 1–5, 9, 10, 17, 19–21, and 26 of the '138 Patent were the subject of IPR2020-01582, in which we entered a Final Written Decision determining that Petitioner did not establish that these claims are unpatentable. *See Netflix, Inc. v. Avago Techs. Int'l Sales Pte. Ltd.*, IPR2020-01582, Paper 32 (PTAB Mar. 11, 2022) ("'1582 Final Written Decision").

*C. The '138 Patent (Ex. 1001)*

The '138 Patent relates to automated allocation and management of computing functions and resources within a distributed computing system. *See* Ex. 1001, 1:11–12, 1:37–42.

Figure 1 of the ’138 Patent is reproduced below.

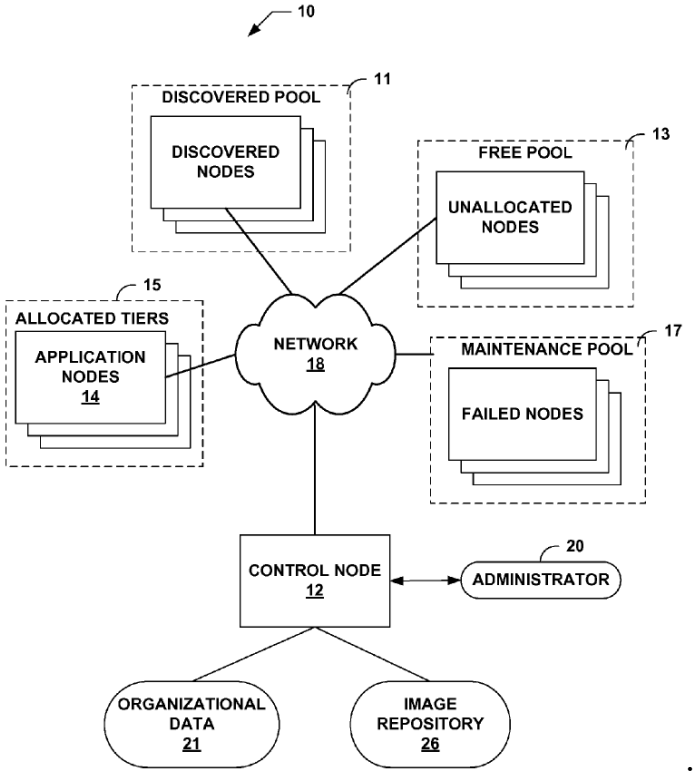


Figure 1 depicts distributed computing system 10 constructed from a collection of computing nodes. *See* Ex. 1001, 2:40–41, 3:54–56. Distributed computing system 10 includes physical computing devices or nodes operating in cooperation with each other to provide distributed processing, including at least one control node 12, and network 18 that permits internode communications. *See id.* at 3:56–58, 3:62–66, 4:60–5:4. Control node 12 maintains organizational data 21, stores software image instances in image repository 26, and interacts with system administrator 20 (a user). *See id.* at 5:23–25, 6:4–6, 13:27–32, Fig. 2. Control node 12 also provides system support functions for distributed computing system 10, including managing the roles of each computing node and the execution of software applications. *See id.* at 4:46–51, 5:12–13, 5:40–42.

The computing nodes are logically grouped within discovered pool 11, free pool 13, allocated tiers 15, and maintenance pool 17. *See* Ex. 1001, 3:59–62. Discovered pool 11 includes a set of computer nodes discovered by control node 12. *See id.* at 4:6–8. Free pool 13 is a set of unallocated nodes that are available for use. *See id.* at 4:26–27. Allocated tiers 15 include one or more tiers of application nodes 14 that currently provide a computing environment for execution of software applications. *See id.* at 4:35–37. Maintenance pool 17 includes a set of computing nodes that could not be inventoried or have been taken out of service. *See id.* at 4:43–45.

Figure 24 of the '138 Patent is reproduced below.

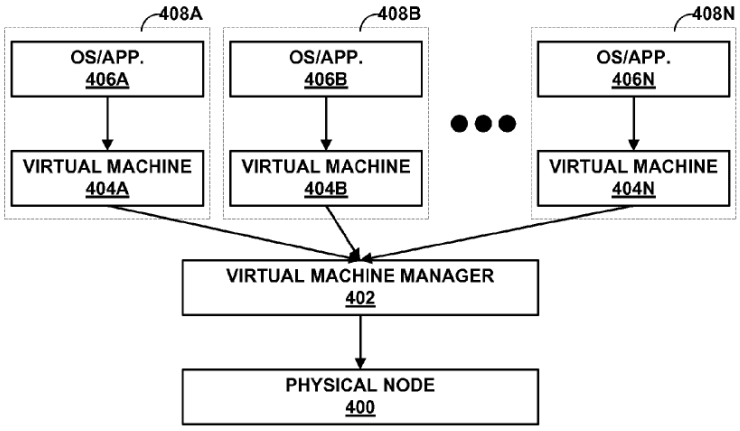


Figure 24 depicts a block diagram of a physical node to which a virtual machine manager and one or more virtual machines are deployed. *See* Ex. 1001, 3:28–30, 32:54–55. Physical node 400 may be any node within distributed computing system 10, for example, any one of application nodes 14. *See id.* at 32:56–58. Virtual machine manager 402 executes on physical node 400 and hosts virtual machines 404A through 404N. *See id.* at 32:59–60, 32:65–67. A virtual machine is "software that creates an environment that emulates a computer platform." *Id.* at 32:67–33:2.

Multiple operating systems and software applications 406A through 406N may execute on virtual machines 404A through 404N, respectively, and virtual machines 404 may execute on a single physical node or multiple physical nodes. *See id.* at 33:6–11. Virtual machine manager 402 utilizes operating system data and application data for execution of each virtual machine 404. *See id.* at 33:25–27. Operating system data, application data, and instance-specific configuration data for the underlying virtual machines 404 are stored as respective virtual disk image files 408. *See id.* at 33:27–30. Virtual machine disk image file 408A provides a bootable software image for operating system/application information 406A that is capable of being executed on virtual machine 404A. *See id.* at 33:35–39.

Figure 21 of the '138 Patent is reproduced below.

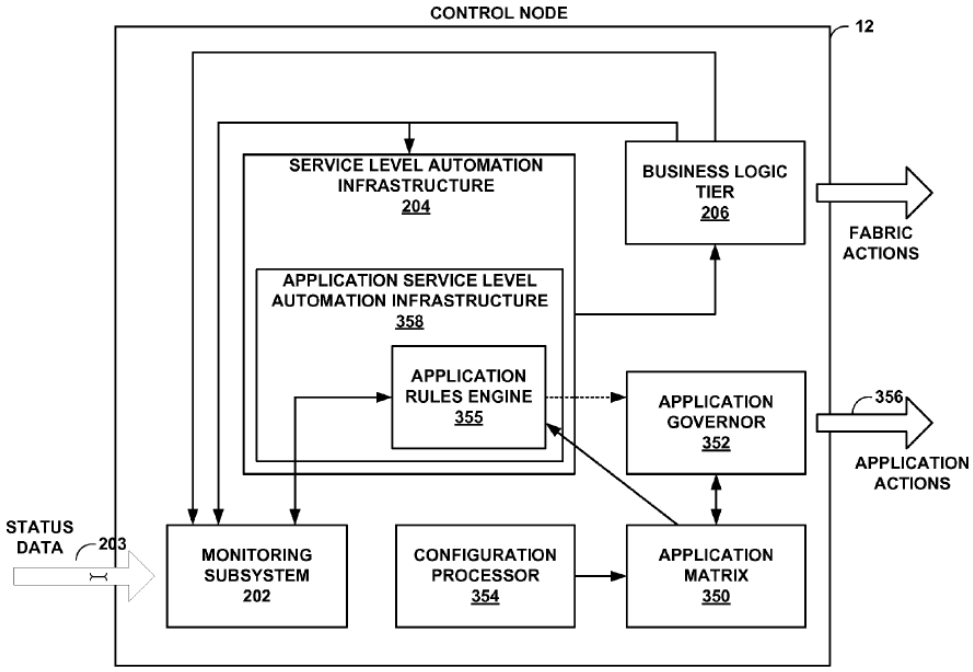


Figure 21 depicts a block diagram of control node 12. *See* Ex. 1001, 3:22–23, 26:62–63. Control node 12 includes monitoring subsystem 202, service level automation infrastructure (SLAI) 204, business logic tier 206,

application matrix 350, application governor 352, configuration processor 354, and application service level automation infrastructure (application SLAI) 358. *See id.* at 3:8–11, 19:53–57, 26:63–67. Application matrix 350, application governor 352, configuration processor 354, and application SLAI 358 provide a framework that allows control node 12 to autonomically control the deployment, execution, and monitoring of applications across applications nodes 14 of distributed computing system 10. *See id.* at 27:1–6. Application matrix 350 contains all the information needed by control node 12 to interact with one or more applications or application servers, and to provide autonomic control over a set of applications. *See id.* at 27:7–10. More specifically, application matrix 350 provides a logical definition for deploying and controlling the set of applications within distributed computing system 10, and may be an electronic document that conforms to a data description language such as XML. *See id.* at 27:10–15.

Application SLAI 358 includes application rules engine 355 dedicated to processing application-level rules to provide autonomic control over the applications defined within application matrix 350. *See* Ex. 1001, 27:15–19. In order to give effect to changes in application matrix 350, application SLAI 358 automatically updates application rules engine 355 and monitoring subsystem 202. *See id.* at 27:21–24. Application SLAI 358 captures configuration attributes and rule attributes contained in application matrix 350 in response to an alert from application matrix 350. *See id.* at 27:24–30, 32:29–35, Fig. 23 (steps 390, 392). Application SLAI 358 transfers any new rule attributes to the working memory of application rules engine 355 to provide autonomic control over the deployment, monitoring,

and execution of the applications defined within application matrix 350. *See id.* at 27:30–34, 32:35–38, 32:42–49, Fig. 23 (step 394).

Application governor 352 is a software engine that performs application-level actions (e.g., deployment and monitoring applications) based on requests from application rules engine 355. *See* Ex. 1001, 27:56–64. Application governor 352 uses application matrix 350 as a source of parameters when carrying out the application-level operations requested by application rules engine 355. *See id.* at 27:65–67. As an example, applications rules engine 355 detects that a node, to which the application is not deployed, is ready for use by the application and directs application governor 352 to handle the details of deploying the application to that node by accessing application matrix 350 to retrieve application-specific parameters necessary to deploy the application. *See id.* at 27:67–28:7. Application governor 352 performs a similar procedure to undeploy an application. *See id.* at 28:12–23.

### *D. Illustrative Claims*

Challenged claims 11, 12, and 14 depend directly or indirectly from claim 1, and challenged claims 22 and 24 depend directly from independent claim 19. Claims 11 and 12 are illustrative of the challenged claims and are reproduced below, along with the claims from which they depend:

> 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.

9. The distributed computing system of claim 1, wherein the control node further comprises one or more rule engines that provide autonomic deployment of the software applications to the virtual machines in accordance with a set of one or more rules.

11. The distributed computing system of claim 9, wherein the automation infrastructure automatically updates the one or more rules engines to autonomically control the deployment of the software applications to the application nodes in accordance with a current state of an application matrix.

12. The distributed computing system of claim 1, wherein the control node further comprises an application matrix that includes parameters for controlling the deployment, undeployment, and execution of the software applications to the virtual machines within the distributed computing system.

*E. Asserted Challenge to Patentability and Asserted Prior Art*

Petitioner asserts that claims 11, 12, 14, 22, and 24 would have been unpatentable on the following ground (Pet. 8):

| Claim(s) Challenged | 35 U.S.C. § | Reference(s) |
|---|---|---|
| 11, 12, 14, 22, 24 | 103[1] | Khandekar[2], Le[3], Matthews[4] |

## II. ANALYSIS

*A. Claim Construction*

In an *inter partes* review proceeding, the Board applies the same claim construction standard as that applied by federal courts in a civil action under 35 U.S.C. § 282(b), which is generally referred to as the *Phillips* standard. *See Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc); 37 C.F.R. § 42.100(b). Under the *Phillips* standard, "words of a claim 'are generally given their ordinary and customary meaning.'" *Phillips*, 415 F.3d at 1312 (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (1996)).

*1. Three-Step Deployment Process*

Patent Owner asserts that each of the challenged claims, by way of their dependence from either claim 1 or independent claim 19, requires an ordered three-step deployment process (or claim elements that perform the ordered three-step deployment process) described in the '138 Patent

[1] The Leahy-Smith America Invents Act, Pub. L. No. 112-29, 125 Stat. 284 (2011), amended 35 U.S.C. § 103 effective March 16, 2013. Because the '138 Patent has an effective filing date prior to the effective date of the applicable AIA amendment, we refer to the pre-AIA version of § 103.
[2] Ex. 1004, US 7,577,722 B1, issued Aug. 18, 2009 ("Khandekar").
[3] Ex. 1005, US 8,209,680 B1, issued June 26, 2012 ("Le").
[4] Ex. 1019, US 2003/0018699 A1, published Jan. 23, 2003 ("Matthews").

Specification. *See* PO Resp. 6, 13–44. Petitioner disagrees with Patent Owner's assertions. *See* Pet. Reply 3–6.

As demonstrated in the analysis below in Section II.D., we need not decide whether the challenged claims require an ordered three-step deployment process, as asserted by Patent Owner.

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

Petitioner asserts that a person of ordinary skill in the art would have understood the limitation "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," as recited in unchallenged independent claims 1 and 19, to be "satisfied by, but not limited to, a plurality of virtual disk image files." Pet. 15. Petitioner points out that unchallenged claim 2, dependent from claim 1, specifies that "the image instances of the plurality of software applications comprise virtual disk image files." *Id.* (quoting Ex. 1001, 36:50–53). According to Petitioner, "[a] teaching sufficient to satisfy claim 2 must also be sufficient to satisfy claim 1." *Id.* (citing Ex. 1003 ¶ 37).

Patent Owner contends that although claim 2 makes clear that "'virtual disk image files' can be a type of 'image instances of the plurality of software applications,'" the language of claim 1, as well as claim 19, requires more. PO Resp. 31–32. Noting the claim language shown here in italics—"a plurality of image instances of a plurality of software applications *that are executable on the plurality of virtual machines*"—Patent Owner argues that Petitioner's position disregards the italicized language by effectively substituting "virtual disk image files" for the entire

limitation. PO Resp. 31–32; *see* PO Sur-reply 6 (quoting '1582 Final Written Decision at 11). Highlighting the difference, Patent Owner argues that, unlike the italicized language in claim 1, claim 2 does not address how or where the virtual disk image files are deployed. PO Resp. 32 (citing Ex. 1001, 36:50–52; Ex. 2001 ¶ 105). Further, according to Patent Owner, the italicized language reflects part of a three-step deployment process of the claimed invention. *Id*. at 32; *see id*. at 13–30 (discussing claim language and Specification support for the purported three-step deployment process). Ultimately, Patent Owner contends that the Board should "apply the plain and ordinary meaning" of the full limitation. *Id.* at 33.

In its Reply,[5] Petitioner disputes Patent Owner's contention that Petitioner's proposed interpretation of this limitation reads Patent Owner's three-step deployment process out of the claims, asserting that "it is the claim language that requires that image instances need only be 'executable'—not actually executed." Pet. Reply 6. Petitioner further asserts that Patent Owner "admits that 'virtual disk image files' can satisfy th[e] limitation." *Id.* (citing PO Resp. 31; Ex. 1024[6], 39:2–20, 81:8–23).

In its Sur-reply, Patent Owner disputes Petitioner's assertion that Patent Owner made such an admission. According to Patent Owner, Dr. Rosenblum's testimony cited by Petitioner "did not even imply that a

[5] In addition to the arguments summarized here, Petitioner presents an alternative claim construction in the Reply. *See* Pet. Reply 15. For the reasons explained below in Section II.D.3.a.4.c., we do not consider Petitioner's Reply arguments advancing a new theory (Pet. Reply 14–16), which exceed the proper scope of a Reply.

[6] Ex. 1024 is Dr. Rosenblum's deposition testimony from IPR2020-01582, also challenging claims of the '138 Patent.

virtual disk image file can satisfy the entirety of this limitation – specifically, the portion requiring 'a plurality of image instances of a plurality of software applications **that are executable on the plurality of virtual machines**.'" PO Sur-reply 6 (quoting Ex. 1001, 36:35–38). Patent Owner contends all that Dr. Rosenblum stated was that image instances of software applications can constitute virtual disk image files, which is a position not in dispute. *See id.* at 5–6 (reproducing Ex. 1024, 81:8–23; citing Ex. 1024, 39:2–20; PO Resp. 31). Patent Owner contends that Dr. Rosenblum rejected Petitioner's position in his Declaration. *See id.* at 6 (quoting Ex. 2001 ¶ 105). Patent Owner further highlights that, in the '1582 Final Written Decision, the Board found that Petitioner's proposed interpretation reads "that are executable on the plurality of virtual machines" out of the claim language. *See id.* (quoting '1582 Final Written Decision at 11) (emphasis omitted).

This limitation was not expressly construed in the Institution Decision in this case (*see* Dec. 10–11) or in the Institution Decision or Final Written Decision in related IPR2020-01582 (*see* IPR2020-01582, Paper 9 at 9–10 (PTAB Mar. 15, 2021); '1582 Final Written Decision at 10–12). Consistent with the '1582 Final Written Decision, we agree with Patent Owner that Petitioner's proposed interpretation, namely, that the limitation is satisfied by "a plurality of virtual disk image files," reads "that are executable on the plurality of virtual machines" out of the claim language. *See* Pet. 15; PO Resp. 31–32. Dependent claim 2, on which Petitioner's argument relies, makes clear that "virtual disk image files" can be "image instances of the plurality of software applications"—but claim 2 does not refer to the additional language "that are executable on the plurality of virtual machines" recited in claim 1 as well as claim 19. Ex. 1001, 36:35–37, 36:50–52,

38:9–11, 39:4–6. Accordingly, we do not adopt Petitioner's proposed interpretation of the limitation.

We again need not expressly construe "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," but we note that "[a] claim construction that gives meaning to all the terms of the claim is preferred over one that does not do so." *Merck & Co., Inc., v. Teva Pharms. USA, Inc.*, 395 F.3d 1364, 1372 (Fed. Cir. 2005); *see also Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 950 (Fed. Cir. 2006) ("[C]laims are interpreted with an eye toward giving effect to all terms in the claim."). In this Decision, the words of the entire limitation, including "that are executable on the plurality of virtual machines," are given their ordinary and customary meaning.

*3. "image instance"*

Both parties apply the construction of "image instance" to which they agreed in the related district court proceeding—specifically, "a snapshot, or copy, of installed and configured software." Ex. 2003 (Joint Claim Construction and Prehearing Statement in the related district court proceeding), 1–2 (identifying this construction of "image instance" as a stipulated claim construction); PO Resp. 42 (citing Ex. 2001 ¶ 137; Ex. 2003, 2) (arguing that Petitioner "admits" to this construction and applying this construction to dispute Petitioner's unpatentability showing); Pet. Reply 16 n.5 (referring to an image instance as "a snapshot, or copy, of installed and configured software"). Because there is no apparent dispute about the proper construction of "image instance" on the record before us, we need not construe this claim term. *Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1375 (Fed. Cir. 2019) ("The Board is required to construe 'only

those terms . . . that are *in controversy*, and only to the extent necessary to resolve the controversy.'" (emphasis added) (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

*4. "virtual machine"*

Patent Owner presents arguments addressing the construction of "virtual machine." *See* PO Resp. 7–13. Petitioner does not address Patent Owner's construction of "virtual machine." *See generally* Pet. Reply. As demonstrated in the analysis below in Section II.D., we need not construe "virtual machine." *See Realtime Data*, 912 F.3d at 1375.

*5. "application matrix"*

Petitioner asserts that the term "application matrix," recited in each of the challenged claims, "is satisfied by, but not necessarily limited to, an encoded, electronic document that includes application-specific parameters for controlling the deployment of applications." Pet. 15–16 (quoting Ex. 1001, 2:55–65, 28:34–38; citing Ex. 1001, 28:34–38, 28:5–11, 31:7–50; Ex. 1003 ¶¶ 39–40). Patent Owner does not address Petitioner's construction of "application matrix." *See* PO Resp. 6; PO Sur-reply 2–5. As demonstrated in the analysis below in Section II.D., we need not construe "application matrix." *See Realtime Data*, 912 F.3d at 1375.

*B. Principles of Law*

A claim is unpatentable under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual

determinations, including (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) if in evidence, so-called secondary considerations. *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17–18 (1966).[7]

"In an [*inter partes* review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable." *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016). Although a burden of production may shift, the burden of persuasion on the issue of patentability always remains with the petitioner and never shifts to the patent owner. *Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015). To prevail, the petitioner must support its challenge by a preponderance of the evidence. 35 U.S.C. § 316(e).

*C. Level of Ordinary Skill in the Art*

Petitioner asserts that a person of ordinary skill in the art of the '138 Patent would have had either (a) "a bachelor's degree in electrical engineering, computer science, or a similar field with at least two years of experience in distributed computing systems"; or (b) "a master's degree in electrical engineering, computer science, or a similar field with a specialization in distributed computing systems." Pet. 13–14 (citing Ex. 1003 ¶¶ 33–34). Petitioner further contends that "[a] person with less formal education but more relevant practical experience may also meet" the

[7] The parties do not present evidence of secondary considerations.

standard of a person of ordinary skill in the art. *Id.* Petitioner also asserts that a person of ordinary skill in the art

> in 2006 with the above level of skill would have had the knowledge and skills necessary to:
> - Create a distributed computing system with a network of computing devices/servers;
> - Use virtualized computing systems, including the use of commercially available software from VMware such as ESX and GSX software;
> - Create images of physical and virtual machines, including images that include [virtual machine monitors (VMMs)[8]], operating systems, and applications;
> - Use rules engines and other knowledge-based systems to automatically deploy images to the nodes of a distributed computing system;
> - Implement multi-tier applications and web services on distributed computer systems; and
> - Implement systems to monitor the status and state of distributed computer systems.

Pet. 14 (citing Ex. 1003 ¶ 35); *see* Ex. 1003 ¶ 12.

Patent Owner contends that of a person of ordinary skill in the art "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." PO Resp. 6 (citing Ex. 2001 ¶ 24). Patent Owner notes the similarities of Petitioner's

[8] Petitioner uses the acronym VMM to refer to the virtual machine monitors disclosed in Khandekar and Le, consistent with the respective use of VMM in Khandekar and Le, and to refer to a virtual machine manager as disclosed and claimed in the '138 Patent. *See, e.g.*, Pet. 7–8; Ex. 1003 ¶ 12; Ex. 1004, 4:67–5:1; Ex. 1005, 12:38. To avoid confusion, throughout the Decision, we use VMM to refer only to a virtual machine monitor and use the full term for the claimed virtual machine manager.

description of a person of ordinary skill in the art, and asserts that "[a]ny differences in the parties' view of a [person of ordinary skill in the art] are not material to the arguments herein." *See id.* at 6–7 (citing Ex. 2001 ¶ 25).

We determine that a person of ordinary skill in the art would have had a bachelor's degree in electrical engineering, computer science, or a similar field with two years of experience in distributed computing systems, or a master's degree in electrical engineering, computer science, or a similar field with a specialization in distributed computing systems. Our adopted level of ordinary skill is consistent with the level of skill advocated by the parties and the level of skill reflected by the '138 Patent Specification and the asserted prior art.

### *D. Patentability Challenge*

#### *1. Overview of Khandekar (Ex. 1004)*

Khandekar is directed to the application of virtual machine (VM) technology, including creating, configuring, and deploying computer systems with user-specified features. *See* Ex. 1004, 1:7–10, 4:30–35. "The main idea of the invention is that a user who wants access to a computer with specific configuration characteristics interacts with [a] provisioning server . . . and, by means of [a] user console[,] . . . selects components of the computer that he wants to have." *Id.* at 9:65–10:3.

Figure 2 of Khandekar is reproduced below.

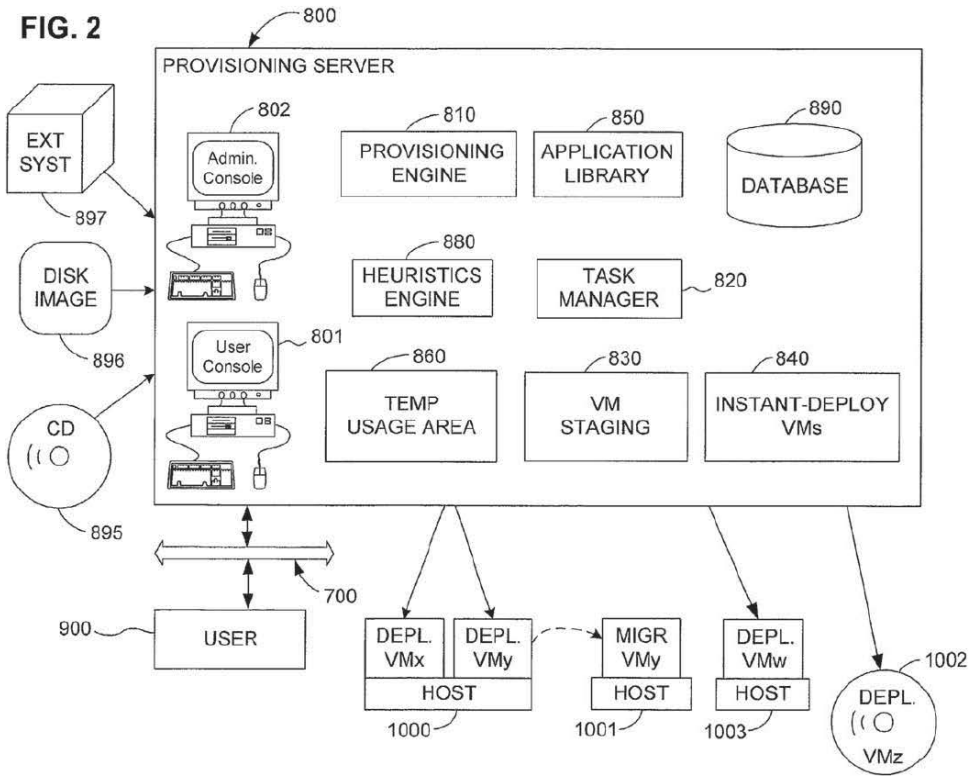


Figure 2 illustrates the main components of VM provisioning. *See* Ex. 1004, 4:17–18, 9:10–12. Provisioning server 800 includes system hardware, system software, and devices as needed. *See id.* at 9:26–32. User console 801 allows end users to interact with provisioning server 800 via network 700 to specify input parameters. *See id.* at 9:46–49, 9:53–56, 13:65–15:39. Administration console 802 allows an administrator to interact with provisioning server 800 to monitor the operational state of the various computers and maintain database 890. *See id.* at 9:59–64, 18:1–55. Heuristics engine 880 takes user input and applies heuristics/rules to select the most appropriate VM from VM staging library 830. *See id.* at 10:27–29, 15:40–17:29, Fig. 3. At least one host computer 1000, 1001 is made available to provisioning server 800, and heuristics engine 880 selects which host 1000, 1001 is most appropriate to provision and on which to deploy a

VM. *See id.* at 10:29–33. Heuristics engine 880 selects the applications from library 850 to install in the newly created VM. *See id.* at 10:33–35, 18:56–19:18, Fig. 5.

Library 840 stores instant deployment VMs kept in a suspended state until deployed, which allows a user to select a fully configured VM according to an instant-deployment embodiment. *See* Ex. 1004, 11:20–24, 17:30–67, Fig. 4. After the user selects a fully configured VM, provisioning server 800 selects an appropriate host 1000, 1001 on which to provision the VM, copies necessary data to that host, and resumes execution of the VM on the host. *See id.* at 11:24–27.

Khandekar discloses that some interface is usually required between a VM and the underlying host platform, particularly between the VM and the host operating system, and this interface is referred to as the virtual machine monitor (VMM). *See* Ex. 1004, 6:63–7:6.

> 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.

Ex. 1004, 13:19–35.

*2. Overview of Le (Ex. 1005)*

Le is directed to the creation, manipulation, and deployment of computer disk images. *See* Ex. 1005, 1:13–14. Le discloses a Universal Computer Management System (UCMS), which is an enhanced disk

imaging system for both physical computers and virtual machines. *See id.* at 59:21–24. The UCMS manages a set of physical computers and virtual machines residing on a network using a common set of image files. *See id.* at 59:29–30, 65:20–22. The UCMS server includes an imaging server, a registration database, and a resource database that may include template image files. *See id.* at 60:6–28, 60:64–70:3, Fig. 6.

Le discloses that when a virtual machine is powered on, a virtual machine monitor (VMM) program controls the virtual machine and manages the interactions between the guest software running inside the virtual machine and the host's physical resources such as hardware devices. *See* Ex. 1005, 65:65–66:2. Some products, such as VMware GSX Server, access the host's physical resources through an industry-standard host operating system such as Linux or Windows. *See id.* at 66:2–6. Other products, like VMwareESX Server, include a VMM and system-level kernel capable of managing physical resources directly without the need for a host operating system. *See id.* at 66:6–10. A virtual machine manager is typically responsible for creating, destroying, and maintaining virtual machine files residing on the host. *See id.* at 66:11–13.

To deploy a template image to a virtual machine, the UCMS specifies a destination virtual machine host and interacts with a virtual machine manager to copy the template image and a virtual machine configuration file to the host. *See* Ex. 1005, 66:33–42, 73:44–47. To create a template image from a virtual machine's virtual disk, the virtual machine files are copied from the virtual machine manager to the UCMS imaging server. *See id.* at 66:65–67:4. A typical template image encapsulates the software stack,

which contains an operating system, one or more applications, and data files. *See id.* at 67:30–40.

*3. Analysis*

*a. Unchallenged Claim 1*

We begin our analysis with claim 1, which is not challenged in this proceeding, because challenged claims 11, 12, and 14 include the limitations of claim 1 by virtue of their dependency. The limitation of claim 1 that recites "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines" is dispositive to our analysis of Petitioner's challenge to claims 11, 12, and 14. Below, we summarize Petitioner's arguments regarding the preamble and preceding limitations of claim 1 for context and then provide our analysis of the dispositive limitation.

*1. "A distributed computing system comprising:"*

Petitioner contends that Khandekar teaches "[a] distributed computing system," as recited in claim 1, based on Khandekar's disclosure of a network of cooperating VMs deployed on respective physical host platforms. *See* Pet. 22 (citing Ex. 1004, code (57), 4:30–5:13 12:40–50, 23:26–24:7, Figs. 2, 7; Ex. 1003 ¶¶ 66–69).

In addition, Petitioner asserts that Le teaches "[a] distributed computing system," as recited in claim 1, based on Le's disclosure of a distributed computing system including a UCMS that plays an analogous role to Khandekar's provisioning server. *See* Pet. 22–23 (quoting Ex. 1005, 59:29–30; citing Ex. 1005, 53:45–57, 60:19–28; Ex. 1003 ¶¶ 70–73; Pet. 18–22).

*2. "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"*

Petitioner asserts that Khandekar teaches the distributed computing system comprises "a software image repository comprising non-transitory computer-readable media operable to store . . . a plurality of image instances," as recited in claim 1, based on Khandekar's disclosure of a provisioning server that includes various libraries for storing software images. *See* Pet. 23 (reproducing Ex. 1004, Fig. 2; quoting Ex. 1004, 4:36–38, 11:37; citing Ex. 1004, 4:38–41, 11:36–49, 23:28–51, 23:54–24:7; Ex. 1003 ¶¶ 74–77). Petitioner contends Khandekar discloses that the provisioning server includes several libraries/repositories for software images and that the provisioning server acts as a central repository with images of virtual machines and applications. *See id.* at 24 (citing Ex. 1004, 11:21–25, 11:50–58, 16:43–63, 18:1–42, 18:56–19:2; Ex. 1003 ¶ 76). According to Petitioner, "Khandekar teaches using standard computer components." *Id.* at 24 (citing Ex. 1004, 9:26–32; Ex. 1003 ¶ 77).

Petitioner further asserts that Khandekar teaches a "virtual machine manager that is executable on a plurality of application nodes" based on Khandekar's disclosure that a VMM is installed on each of a plurality of hardware hosts. *See* Pet. 25 (quoting Ex. 1004, 4:66–5:2, 7:7–9; citing Ex. 1004, 5:2–8, 7:10–27, 13:17–35; Ex. 1003 ¶¶ 81–94). Petitioner contends Khandekar teaches that the provisioning server stores a plurality of image instances of VMs and that when the provisioning server provides a VM to a host, it may also include a VMM together with the VM. *See id.* at 25–26 (quoting Ex. 1004, 8:41–44, 13:17–28; citing Ex. 1004, 8:36–41,

13:28–35; Ex. 1003 ¶ 82; Pet. 36–41). Petitioner asserts that a person of ordinary skill would have found it obvious to store VMM images in a single repository in the provisioning server because Khandekar teaches storing VMs there, and a corresponding VMM is included and installed together for each VM. *See id.* at 24 (citing, e.g., Ex. 1004, 8:36–44, 13:17–35; Ex. 1003 ¶ 75).

Petitioner asserts that it would have been obvious to one of ordinary skill in the art to modify Khandekar's teachings to include storing a plurality of VMM images, in addition to the VM images, in the provisioning server, in order to deploy the VMMs and VMs together. *See* Pet. 24, 26–29; *KSR*, 550 U.S. at 418. According to Petitioner, "[s]ince VM images are stored in the provisioning server, a [person of ordinary skill in the art] 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." Pet. 26 (citing Ex. 1004, 8:36–44, 13:17–35); *see id.* at 24 (citing Ex. 1004, 8:36–44, 13:17–35; Ex. 1003 ¶ 75; Pet. 25–33), 26–27 (citing Ex. 1004, 4:36–41, 8:36–44, 11:21–25, 13:17–35, 16:43–63; Ex. 1003 ¶¶ 48, 84). Petitioner further contends that "[d]oing so would have allowed the hosts to be remotely initialized with VMM software and configured to support [VMs], facilitating Khandekar's goals of automated, flexible deployment [of VMs to hosts] and removing the need to manually install VMMs on hosts." *Id.* at 26 (citing Ex. 1004, 4:30–35; Ex. 1003 ¶ 83).

Petitioner contends that, in addition to Khandekar, Le also teaches the distributed computing system comprises "a software image repository comprising non-transitory, computer-readable media," as recited in claim 1, based on Le's disclosure of a server computer that includes a server disk that

stores image files and template images. *See* Pet. 24 (citing Ex. 1005, 25:34–41, 26:6–22, 26:38–51, 26:63–27:9, 60:64–61:8, 63:20–34, 66:66–67:29, Figs. 4–6; Ex. 1003 ¶¶ 78–80; Pet. 18–22). Petitioner further asserts that Le teaches "a virtual machine manager that is executable on a plurality of application nodes," as recited in claim 1, based on Le's disclosure of a VMM that controls and manages interactions between software running inside the VM and the host's physical resources, and disclosure of the use of VMware ESX software. *See id.* at 29–33 (citing Ex. 1005, 13:22–30, 65:65–66:64, 67:9–32; Ex. 1003 ¶¶ 95–98; Pet. 18–22).

With respect to claim 1's requirement that the software image repository be "operable to store . . . a plurality of image instances of a virtual machine manager," Petitioner asserts that Le's teachings "complement and elaborate on Khandekar's teachings, further rendering it obvious for a plurality of VMM images to be stored in [Khandekar's] provisioning server." Pet. 30 (citing Ex. 1003 ¶¶ 99–111; Pet. 18–22). Specifically, Petitioner contends that Le teaches implementation details of image creation and management for distributed computing and VM systems, and that these teachings address two functions raised by Khandekar—allowing a VMM to be provided from the provisioning server to hosts and allowing the provisioning server to know the characteristics of the host. *See id.* at 19 (citing Ex. 1003 ¶ 59). According to Petitioner, "Le teaches techniques for imaging the hard disk of a remote computer to deploy a cloned image with an operating system (e.g., Type 1 VMM) and applications (e.g., Type 2 VMM)." *Id.* at 20 (citing Ex. 1005, 1:64–3:13, 2:19–21, 15:63–65, 16:29–30, 54:14–21; Pet. 25–32, 42–47); *see also id.* at 30–31 (explaining in more detail Le's disk imaging) (citations omitted). Petitioner further asserts

that Le teaches storing a plurality of VMM images and teaches a UCMS, which is analogous to Khandekar's provisioning server. *See id.* at 31–32 (quoting Ex. 1005, 26:55–56, 54:19–21, 67:31–34; citing Ex. 1005, 26:6–22, 26:53–62, 54:14–19, 59:21–27, 65:20–33, 66:66–67:8, 67:25–30, 67:34–40, Figs. 4–6, 9; Ex. 1003 ¶¶ 104–105; Pet. 18–22). Petitioner contends Le teaches that its system determines the hardware characteristics of the remote computer. *See id.* at 21 (citing Ex. 1005, 70:13–35; Ex. 1003 ¶ 59). According to Petitioner, Le also explains that it was common to create one image file per family of similar computers. *See id.* at 32 (citing Ex. 1005, 9:66–10:9, 16:29–41; Ex. 1003 ¶ 106).

Petitioner contends that it would have been obvious to one of ordinary skill in the art to modify Khandekar's teachings to include storing a plurality of VMM images in the provisioning server, as taught by Le, in order to (1) account for different computer hardware and software platforms for the host computers, (2) provide an appropriate VMM along with the VM to a host computer, and (3) allow rapid deployment of disk images including the operating system and software applications such as Type 1 and Type 2 VMMs. *See* Pet. 19 (citing Ex. 1003 ¶ 59), 30–31 (citing Ex. 1003 ¶ 102), 32 (citing Ex. 1003 ¶ 105).

*3. "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"*

Petitioner asserts that Khandekar teaches "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," as recited in claim 1, based on Khandekar's disclosure in Figure 1 of VMMs that are executed on a host computer and provide a plurality of VMs. *See* Pet. 33–34 (reproducing Ex. 1004, Fig. 1; quoting Ex. 1004, 5:51–58; citing Ex. 1004, 7:7–27, 13:17–35; Ex. 1003 ¶¶ 112–116). Petitioner further contends that Khandekar teaches each of the VMs is operable to provide an environment that emulates a computer platform. *See id.* at 35–36 (quoting Ex. 1004, 6:23–43, 7:45–47; citing Ex. 1004, 6:44–61, 7:39–44; Ex. 1003 ¶ 115). Petitioner asserts that, in addition to Khandekar, Le teaches this limitation of claim 1, based on Le's disclosure of a single VMM that supports a plurality of VMs on a host. *See id.* at 35 (quoting Ex. 1005, 65:43–50; citing Ex. 1005, 65:65–66:10, Fig. 8; Ex. 1011, 1, 5, Fig. 1; Ex. 1016 ¶ 76; Ex. 1003 ¶¶ 117–121; Pet. 18–22).

*4. "(ii) a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines"*

The Petition and Reply offer several different theories to explain how Khandekar allegedly teaches or suggests "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," as recited in claim 1. We address each theory in turn.

*a) First Theory*

Petitioner asserts that Khandekar teaches the software image repository is operable to store "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," based on Khandekar's teachings of three choices for storing and deploying image instances: pre-built, custom-built, and instantly deployable VMs. *See* Pet. 36–37 (quoting Ex. 1004, 9:10–25; citing Ex. 1004, 10:15–26, 14:4–8, 18:56–19:2; Ex. 1003 ¶¶ 122–139; Pet. 17–18, 47–51).

According to Petitioner, "[e]ach teaching independently renders [this] limitation . . . obvious by teaching a plurality of virtual disk image files." *Id.* at 36.

Petitioner further contends that "[f]or pre-built and instantly[ ] deployable VMs, Khandekar teaches that each virtual disk image file includes an operating system and application data," and "[f]or custom-built VMs, the virtual disk image [file] only includes the operating system." Pet. 36–37 (citing Ex. 1003 ¶ 123), 38–39 (citing Ex. 1004, 15:41–46); *see id.* at 37 (quoting Ex. 1004, 15:43–45; citing Ex. 1004, 5:25–36, 8:50–55, 10:15–26, 14:19–25, 15:40–16:12, 18:16–23, Fig. 3; Ex. 1003 ¶¶ 126, 129), 41 (citing Ex. 1004, 16:43–63; Ex 1003 ¶ 134). Petitioner asserts Khandekar teaches that its provisioning server includes a plurality of images of pre-built VMs, custom-built VMs, and instantly deployable VMs that are stored in a library or repository of the provisioning server. *See id.* at 37 (quoting Ex. 1004, 10:29, 18:16–23; citing Ex. 1004, 10:27–29, 16:43–63, 22:17–22, 23:54–24:27, Figs. 2–3; Ex. 1003 ¶ 126), 37–38 (citing Ex. 1004, 15:29–40, 16:43–52; Ex. 1003 ¶ 128), 38 (citing Ex. 1004, 16:43–17:13). Finally, Petitioner contends that "Khandekar teaches the image instances of a plurality of software applications that are executable on the plurality of virtual machines" on the basis that "each pre-built, custom-built, and instantly[ ]deployable VM includes a virtual disk image file with an OS [(operating system)] and applications that are executable on the virtual machines." *Id.* at 41 (citing Ex. 1004, 6:43–56; Ex. 1003 ¶ 137). Petitioner does not rely on the teachings of Le and/or Matthews to address this limitation. *See id.* at 36–41.

In response, Patent Owner contends that Petitioner relies on the VMs disclosed in Khandekar to satisfy both the recited "image instances of the plurality of software applications" and "virtual machine." *See* PO Resp. 47 (citing Pet. 36–41). Patent Owner asserts that "none of Khandekar's disclosures of pre-built, custom-built, and instantly deployable VMs satisfy" the instant limitation. *Id.* at 42.

In its Reply, Petitioner reiterates its theory by asserting that the "claim language . . . requires that *image instances* . . . be 'executable.'" Pet. Reply 6 (italics added). Moreover, Petitioner confirms that "[f]or software application images (limitation [1e]), Petitioner relied on Khandekar's VM images," i.e., virtual disk image files of pre-built, custom-built, and instantly deployable VMs. *Id.* at 12 (citing Pet. 36–41). Petitioner further asserts that "[f]or virtual machines (limitation 1[d]), Petitioner relied on VMs provided by VMware software," i.e., VMs provided by VMMs. *Id.* (citing Pet. 33–36–41; Ex. 1004, Fig. 1). In addition, consistent with the agreed-upon construction for "image instance," Petitioner asserts that "Khandekar's three types of VM images are 'a snapshot, or copy, of installed and configured software.'" *See* Pet. Reply 16 n.5; Ex. 2003, 2.

In its Sur-reply, Patent Owner asserts that it never disputed that Khandekar teaches software application images. *See* PO Sur-reply 12 (citing Pet. Reply 12). Rather, Patent Owner "disputes that either Khandekar and/or Le disclose 'a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines' as required by the unchallenged Claims 1 and 19." *Id.* (citing Ex. 1001, 36:35–37, 38:9–11).

We agree with Patent Owner that Petitioner makes an inadequate showing for this limitation. Petitioner does not present argument or evidence sufficient to demonstrate that Khandekar's disclosures of virtual disk image files for pre-built, custom-built, and instantly deployable VMs teach or suggest "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines." In particular, giving the words of the entire limitation their ordinary and customary meaning, Petitioner does not present argument or evidence sufficient to demonstrate that Khandekar's VM images or virtual disk image files for pre-built, custom-built, and instantly deployable VMs are "executable on the plurality of virtual machines" (i.e., capable of execution on the plurality of virtual machines).

At best, Petitioner cites Khandekar's teachings that it is the operating system and applications of Khandekar's virtual disk image files—not Khandekar's virtual disk image files themselves—that are executable on the virtual machines. *See* Pet. 41 (citing Ex. 1004, 6:42–56 (disclosing applications installed and loaded for running on the VM); Ex. 1003 ¶ 137). But this is inconsistent with Petitioner's position that it is Khandekar's virtual disk image files themselves that teach "a plurality of image instances of a plurality of software applications." *See id.* at 36–37; *see also* Pet. Reply 12 (confirming Petitioner's position (citing Pet. 36–41)).

For the foregoing reasons, Petitioner does not demonstrate that Khandekar teaches, suggests, or renders obvious "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," as recited in claim 1, under Petitioner's first theory.

*b) Second Theory*

In an alternative position presented in the Petition, Petitioner asserts that "[t]o the extent P[atent Owner] contends the software applications are somehow distinct from the virtual disk image file, Khandekar teaches custom-built VMs where application images (installation files) are stored in an application library and copied to hosts." Pet. 37 (citing Ex. 1003 ¶ 124). More specifically, Petitioner contends the custom-built VMs include VM disk files that are stored in the provisioning server and copied to hosts. *See id.* at 38 (citing Ex. 1004, 16:43–17:13). Petitioner contends that for custom-built VMs, the virtual disk image only includes the operating system. *See id.* at 38–39 (citing Ex. 1004, 15:41–46). According to Petitioner, "[a]pplication images (application installation files) are stored in the provisioning server's application library and copied to hosts." *Id.* at 39 (quoting Ex. 1004, 20:18–20, 20:26–28; citing Ex. 1004, 4:42–51, 19:20–20:30, 21:25–64, 23:28–51; Ex. 1003 ¶ 132). Petitioner asserts that "Khandekar teaches images of software applications stored in an applications library . . . that are executable on VMs." *Id.* (reproducing Ex. 1004, Fig. 5; quoting Ex. 1004, 18:56–61, 11:50–53; citing Ex. 1004, 10:27–42, 11:53–58, 12:13–23, 18:56–9:2; Ex. 1003 ¶ 131).

In response, Patent Owner contends that Khandekar teaches that custom-built VMs will be stored with just their operating systems installed, and further teaches "the use of an 'executable installation file,' *i.e.*, the file used to install an application for the first time." PO Resp. 41–42 (citing Ex. 1004, 15:45–46, 18:61–19:2). According to Patent Owner, "[t]his installation file . . . is not equivalent to the 'image instances of software applications' required by the '138 Patent." *Id.* at 42 (citing Ex. 2001 ¶ 137).

Patent Owner contends that Petitioner "admits that an 'image instance' as used in the '138 Patent refers to 'a snapshot, or copy, of installed and configured software.'" *Id.* (quoting Ex. 2003, 2); *accord* Pet. Reply 16 n.5. Patent Owner asserts that Khandekar's executable installation file is completely different from an image instance, and a person of ordinary skill in the art would not recognize them to be equivalent. *See* PO Resp. 42 (citing Ex. 2001 ¶ 137). Petitioner's Reply does not specifically address Patent Owner's contentions disputing whether Khandekar's application installation files for custom-built VMs teach or suggest "image instances of a plurality of software applications." *See, e.g.*, Pet. Reply 12–13.

We agree with Patent Owner that Khandekar's disclosure of application installation files does not teach or suggest "a plurality of image instances of a plurality of software applications," in accordance with Petitioner's construction of "image instance." *See* PO Resp. 41–42; Pet. Reply 16 n.5 ("Khandekar's three types of VM images are 'a snapshot, or copy, of installed and configured software' . . . ."). We agree with and credit Dr. Rosenblum's persuasive testimony distinguishing Khandekar's "executable installation file" from the recited "image instances of software applications." *See* Ex. 2001 ¶ 137. Accordingly, Petitioner does not present sufficient argument or evidence to demonstrate that Khandekar's application installation files teach or suggest "a plurality of image instances of a plurality of software applications," in accordance with Petitioner's construction of an "image instance" as "a snapshot, or copy, of installed and configured software."

For the foregoing reasons, Petitioner does not demonstrate that Khandekar teaches, suggests, or renders obvious "a plurality of image

instances of a plurality of software applications that are executable on the plurality of virtual machines," as recited in claim 1, under Petitioner's second theory.

*c) Third Theory*

In its Reply, Petitioner offers a third theory to address the limitation "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines." Pet. Reply 14–16. Petitioner contends that the '1582 Final Written Decision incorrectly interpreted "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines." *See id.* at 14 n.4; *accord* Tr. 5:8–12, 55:20–56:1. Petitioner asserts that "[c]laim 1 reflects the ['138 Patent] specification: 'image instances of a plurality of *software applications that are executable* on the plurality of virtual machines.'" Pet. Reply 15. Petitioner further contends "[t]he Petition explained how each of Khandekar's three types of VM images—pre-built, custom, and instantly[ ]deployable—comprise a virtual disk image file with an OS and applications that are executable on the virtual machines." *Id.* at 15–16 (quoting Ex. 1004, 15:43–45, 15:49–52; citing Pet. 37–41; Ex. 1004, 5:25–36, 8:50–55, 10:15–26, 15:40–16:12, 17:30–50, Fig. 3; Ex. 1003 ¶¶ 126, 129, 132).

In response, Patent Owner contends that the section of the Reply titled "Reply to PO's Claim 1 Arguments"—which contains the arguments summarized in the previous paragraph—does not identify arguments in Patent Owner's Response that Petitioner intends to rebut. *See* PO Sur-reply 12–13 (citing Pet. Reply 14 & n.4). Based on Petitioner's assertion that the '1582 Final Written Decision interpreted the claim

limitation incorrectly, Patent Owner contends that this section of the Reply is intended more as a rebuttal to the '1582 Final Written Decision than Patent Owner's Response. *See id.* at 12–13; *accord* Tr. 20:16–21:13. According to Patent Owner, rather than attempting to rebut Patent Owner's position directly, Petitioner attempts to rewrite its Petition, which "does not have any different disclosures with respect to elements [1d] (which first defines 'virtual machines') and [1e] (which first defines 'image instances of a plurality of software applications that are executable on the plurality of virtual machines') of unchallenged Independent Claim 1 than the Petition in IPR2020-01582." PO Sur-reply 15–16 (*comparing* Pet. 33–41, *with* IPR2020-01582 Paper 2 at 37–48).

We agree with Patent Owner that Petitioner's Reply arguments directed to Petitioner's third theory are not responsive to the Patent Owner Response, but instead serve as an attempt to rebut the '1582 Final Written Decision and to rewrite the Petition. Contrary to Petitioner's assertions in its Reply, the Petition did not adequately articulate or explain this theory, which relies on the operating system and applications of Khandekar's VM images or virtual disk image files—as opposed to the VM images or virtual disk image files themselves—to satisfy the "that are executable on the virtual machines" claim language. *See* Pet. Reply 15. The foundation on which Petitioner's Reply argument stands, i.e., "[c]laim 1 reflects the ['138 Patent] specification: 'image instances of a plurality of *software applications that are executable* on the plurality of virtual machines" (Pet. Reply 15), was not

clearly offered as a claim construction in the Petition.[9] *See* Pet. 15. Instead, the explicit claim construction offered in the Petition for "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines" (*see id.*) reads "that are executable on the plurality of virtual machines" out of the claim language, as discussed above in Section II.A.2.

Moreover, as explained above addressing Petitioner's first theory, the Petition, at best, includes a *citation* to Khandekar's teachings that it is the operating system and applications of Khandekar's virtual disk image files that are executable on the virtual machines—without explanation or elaboration of Petitioner's third theory. *See* Pet. 41 (citing Ex. 1004, 6:42–56; Ex. 1003 ¶ 137). Petitioner's Reply also points to citations to Khandekar's disclosure of VM images stored with operating systems and applications installed. *See* Pet. Reply 15 (quoting Ex. 1005, 15:43–45, 15:51–52). These citations to Khandekar, similar to arguments in the Petition, merely highlight that Khandekar's virtual disk image files include an operating system and application data installed. *See, e.g.*, Pet. 36–37 ("For pre-built and instantly[ ]deployable VMs, Khandekar teaches that each virtual disk image file includes an operating system and application data." (citing Ex. 1003 ¶ 123)), 37 ("Pre-built VMs also include operating systems and application data installed." (quoting Ex. 1004, 15:43–45, 15:49–52;

[9] This claim construction position, which Petitioner first articulates in the unpatentability analysis of its Reply, also directly contradicts Petitioner's explicit argument in the claim construction section of the same brief, where Petitioner takes the position that "it is the claim language that requires that *image instances* need only be 'executable.'" Pet. Reply 6 (italics added).

citing Ex. 1004, 5:25–36, 8:50–55, 10:15–26, 14:19–25; 18:16–23, Fig. 3; Ex. 1003 ¶¶ 126, 129, 132)), 38–39 ("For custom-built VMs, the virtual disk image only includes the operating system." (citing Ex. 1001, 15:41–46)).

In sum, Petitioner's Reply arguments (Pet. Reply 14–16) amount to a new theory distinct from the first and second theories offered in the Petition. Although the Petition included a citation to Khandekar's disclosure of applications installed and loaded for running on the VM (*see* Pet. 41 (citing Ex. 1004, 6:42–56; Ex. 1003 ¶ 137)), the arguments presented in the Petition do not explain Petitioner's third theory, elucidated for the first time in the Reply. It is improper for Petitioner on Reply to point to vague, isolated arguments and citations to the prior art from the Petition and advance a new theory that was not explicitly offered or explained in the Petition. Permitting a petitioner to advance a new theory on reply in this manner would put the burden on a patent owner to deduce and respond to all possible theories that could have been, but were not, explicitly raised in a petition. But this would be an unfair burden to place on a patent owner—and one that governing law and Board regulations and guidelines make clear the patent owner does not bear. *See, e.g.*, 35 U.S.C. § 312(a)(3); 37 C.F.R. §§ 42.22(a)(2), 42.104(b)(4)–(5); Consolidated Trial Practice Guide (Nov. 2019) ("CTPG")[10] 73–75. Rather, it is a petitioner's burden to explain with particularity, at the outset in the petition, why the patent it challenges is unpatentable. *Harmonic*, 815 F.3d at 1363; 35 U.S.C. § 312(a)(3). As the Federal Circuit has explained,

> Unlike district court litigation—where parties have greater freedom to revise and develop their arguments over time and in

[10] Available at https://www.uspto.gov/TrialPracticeGuideConsolidated.

> response to newly discovered material—the expedited nature of IPRs bring[s] with it an obligation for petitioners to make their case in their petition to institute. While the Board's requirements are strict ones, they are requirements of which petitioners are aware when they seek to institute an IPR.

*Intelligent Bio-Systems, Inc. v. Illumina Cambridge Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016). In the course of an *inter partes* review, a petitioner may, in certain circumstances, introduce new evidence, *Genzyme Therapeutic Prods. LP v. Biomarin Pharm. Inc.*, 825 F.3d 1360, 1366 (Fed. Cir. 2016), but "[s]hifting arguments [to a new theory absent from the petition] is foreclosed by statute, [Federal Circuit] precedent, and Board guidelines," *Wasica Fin. GmbH v. Cont'l Auto. Sys., Inc.*, 853 F.3d 1272, 1286 (Fed. Cir. 2017).

In sum, the Petition did not adequately articulate or explain Petitioner's third theory as to how Khandekar teaches, suggests, or renders obvious "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," as recited in claim 1. The portion of the Reply newly advancing and supporting Petitioner's third theory (Pet. Reply 14–16) exceeds the proper scope of a Reply and is not considered in rendering this Final Written Decision. *See* 37 C.F.R. § 42.23(b) ("A reply may only respond to arguments raised in the corresponding . . . patent owner response . . . ."); CTPG at 73 ("Petitioner may not submit new evidence or argument in reply that it could have presented earlier . . . ."), 74 ("[A] reply . . . that raises a new issue . . . may not be considered.").

*d) Fourth Theory*

In its Reply, Petitioner offers yet another theory to address the limitation "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines." Pet. Reply 17–19. Petitioner contends that "[t]o the extent the images–rather than the software applications–must be executable on the virtual machines, virtual disk image files are by their nature capable of being booted/run on virtual machines." *Id.* at 17 (quoting Ex. 1001, 33:32–35, 35:59–67). According to Petitioner, "Khandekar teaches virtual disk image files—using the same VMware software as the '138 patent—that are capable of being run/booted on virtual machines provided by VMware ESX software." *Id.* at 18. Petitioner asserts that "[t]he Petition explained how instantly[ ]deployable VM images are bootable on VMs and 'resumed in a fully[ ]booted state.'" *Id.* (citing Pet. 40–41, 50–51; Ex. 1029 ¶ 17). Petitioner asserts that "[f]or images of pre-built and custom-built VMs, Petitioner explained how Khandekar's deployment process includes copying the VM disk files and cited to the description in Khandekar that includes booting the VM images." *Id.* at 18–19 (citing Pet. 48–50; Ex. 1003 ¶¶ 158–161). Petitioner further asserts that "Khandekar teaches deploying VM images on hosts; . . . when executed, the booted VM images run on the VMs." *Id.* at 19 (quoting Ex. 1004, 13:36–40, 19:33–45, 22:53–67; citing Pet. 47–51; Ex. 1004, 8:64–67; Ex. 1029 ¶ 18; Ex. 1003 ¶ 146).

In response, Patent Owner contends that Petitioner's Reply arguments under the heading "Bootable Images" (Pet. Reply 17–19), summarized in the previous paragraph, are entirely new and were not set forth in the Petition. *See* PO Sur-reply 18. Patent Owner contends that the term "'bootable

image' is not found anywhere in the Petition" and "[t]he term 'boot' appears [only] once in the Petition." *Id.* (quoting Pet. 40).

We agree with Patent Owner that Petitioner's fourth theory, namely, its Reply argument that Khandekar teaches virtual disk image files capable of being booted/run on virtual machines, was not adequately presented or explained in the Petition. *Compare* Pet. Reply 17–19, *with* Pet. 36–41, 47–51. Similar to the Reply arguments addressed above, these arguments are not responsive to the Patent Owner Response, but instead serve as an attempt to rebut the '1582 Final Written Decision and to rewrite the Petition.

We are not persuaded by Petitioner's position that "[t]he Petition *explained* how instantly[ ]deployable VM images are bootable on VMs and 'resumed in a fully[ ]booted state.'" Pet. Reply 18 (emphasis added) (citing Pet. 40–41, 50–51; Ex. 1029 ¶ 17). The Petition not only failed to explain Khandekar's teaching of this functionality in any detail, it also did not clearly articulate how this particular functionality allegedly satisfies the "executable on the plurality of virtual machines" claim language. Pet. 40–41. The Reply identifies only a single undeveloped statement in the Petition: "instantly[ ]deployable VMs are suspended and resumed in a fully[ ]booted state." *Id.* (citing Ex. 1004, 17:30–50; citing Ex. 1003 ¶ 135); *see* Pet. Reply 18. The Reply otherwise reproduces quotations from Khandekar's disclosure that were either cited or quoted in the Petition. *Compare* Pet. Reply 18–19 (quoting Ex. 1004, 17:41–43, 11:24–27, 22:11), *with* Pet. 40–41 (citing Ex. 1004 17:30–20), 50–51 (quoting Ex. 1004, 11:20–35, 22:4–5; citing Ex. 1004 21:65–22:13).

Like Petitioner's third theory newly presented in the Reply, Petitioner's Reply arguments regarding Khandekar's virtual disk image files

capable of being booted/run on virtual machines amount to new theory distinct from the first and second theories offered in the Petition. Although the Petition stated that "instantly[ ]deployable VMs are suspended and resumed in a fully[ ]booted state" (Pet. 40–41), this statement does not explain Petitioner's fourth theory, elucidated for the first time in the Reply. As explained above, it is improper for Petitioner to advance a new theory in the Reply that was not explicitly offered or explained in the Petition. *See Wasica*, 853 F.3d at 1286. Petitioner bore the burden to explain with particularity, at the outset in the Petition, why claim 1 is unpatentable. *See Harmonic*, 815 F.3d at 1363.

We conclude that the Petition did not adequately articulate or explain Petitioner's fourth theory as to how Khandekar teaches, suggests, or renders obvious "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," as recited in claim 1. The portion of Petitioner's Reply newly advancing this fourth theory (Pet. Reply 17–19) exceeds the proper scope of a Reply and is not considered in rendering this Final Written Decision. *See* 37 C.F.R. § 42.23(b); CTPG 73–74.

*e) Conclusion*

For the reasons given above, we determine that Petitioner does not show that Khandekar teaches, suggests, or renders obvious "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," as recited in claim 1.

*b. Unchallenged Independent Claim 19*

Independent claim 19 recites a method with limitations similar to those recited in claim 1. *Compare* Ex. 1001, 38:1–21 (claim 19), *with id.* at

36:26–49 (claim 1). Identical to claim 1, claim 19 recites "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines." *Id.* at 38:9–11. This limitation is dispositive to our analysis of Petitioner's challenge to claims 22 and 24, which depend from claim 19. Petitioner asserts that the combination of Khandekar and Le teaches, suggests, or renders obvious the limitations of claim 19, relying primarily on its arguments addressing claim 1. *See* Pet. 52–55 (citations omitted).

For the same reasons as those explained in Section II.D.3.a.4. addressing claim 1, Petitioner does not demonstrate that Khandekar teaches, suggests, or renders obvious "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," as recited in claim 19.

*c. Challenged Claims 11, 12, 14, 22, and 24*

For its challenge to claims 11, 12, 14, 22, and 24, Petitioner relies on the teachings of Khandekar, Le, and Matthews to address the limitations of these claims, each of which depends ultimately from either unchallenged independent claim 1 or 19. *See* Pet. 59–75. Petitioner does not rely on the teachings of Le and/or Matthews to address the dispositive limitation, "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," of claims 1 and 19. *See id*. For the reasons explained above in our analysis of unchallenged claims 1 and 19 in Sections II.D.3.a.4. and II.D.3.b., Petitioner does not demonstrate that Khandekar teaches, suggests, or renders obvious "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines," as recited in claims 1 and 19.

After having analyzed the entirety of the record and assigning appropriate weight to the cited supporting evidence, we determine that Petitioner has not established by a preponderance of the evidence that dependent claims 11, 12, 14, 22, and 24 are unpatentable under 35 U.S.C. § 103 as obvious over Khandekar, Le, and Matthews.

## III. CONCLUSION

For the reasons explained above, we conclude as follows:

| Claim(s) | 35 U.S.C. § | Reference(s)/Basis | Claim(s) Shown Unpatentable | Claim(s) Not Shown Unpatentable |
|---|---|---|---|---|
| 11, 12, 14, 22, 24 | 103 | Khandekar, Le, Matthews | | 11, 12, 14, 22, 24 |
| **Overall Outcome** | | | | 11, 12, 14, 22, 24 |

## IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that Petitioner has not shown by a preponderance of the evidence that claims 11, 12, 14, 22, and 24 of the ’138 Patent are unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Harper Batts
Chris Ponder
Jeffrey Liang
SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
hbatts@sheppardmullin.com
cponder@sheppardmullin.com
jliang@sheppardmullin.com

PATENT OWNER:

Daniel S. Young
Chad E. King
ADSERO IP LLC d/b/a SWANSON & BRATSCHUN LLC
dyoung@adseroip.com
chad@adseroip.com