

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.

IPR2020-01582
Patent 8,572,138 B2

Before KRISTEN L. DROESCH, KERRY BEGLEY, and
MELISSA A. HAAPALA, *Administrative Patent Judges*.

DROESCH, *Administrative Patent Judge*.

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

I. INTRODUCTION

A. Background

Netflix, Inc. (“Petitioner”) filed a Petition requesting an *inter partes* review of claims 1–5, 9, 10, 17, 19–21, and 26 (“challenged claims”) of U.S. Patent No. 8,572,138 B2 (Ex. 1001, “’138 Patent”). Paper 2 (“Pet”). Petitioner filed a Declaration of Prashant Shenoy, Ph.D., with its Petition. Ex. 1003. Avago Technologies International Sales Pte. Limited (“Patent Owner”) filed a Preliminary Response. Paper 7 (“Prelim. Resp.”).

We have authority to determine whether to institute review under 35 U.S.C. § 314 and 37 C.F.R. § 42.4(a). An *inter partes* review may not be instituted unless it is determined that “the information presented in the petition filed under section 311 and any response filed under section 313 shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a) (2018).

For the reasons provided below, we determine, based on the record before us, that there is a reasonable likelihood Petitioner would prevail in showing at least one of the challenged claims is unpatentable.

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.), 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. 88; Paper 6, 1.

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.

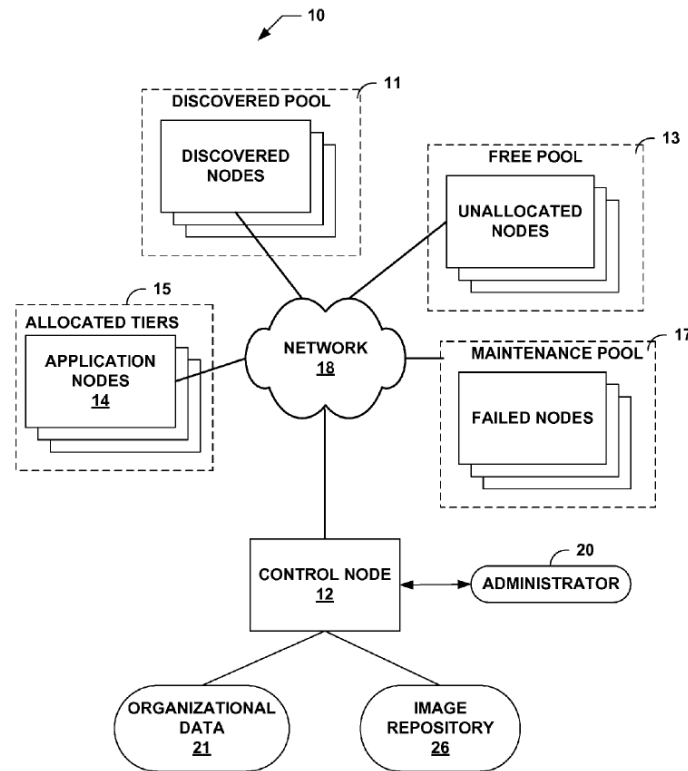


Figure 1 depicts distributed computing system 10 constructed from a collection of computing nodes. *See* Ex. 1001, 2:40–41, 3:54–56.

Distributed computing system 10 includes physical computing devices or nodes operating in cooperation with each other to provide distributed processing, 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 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.

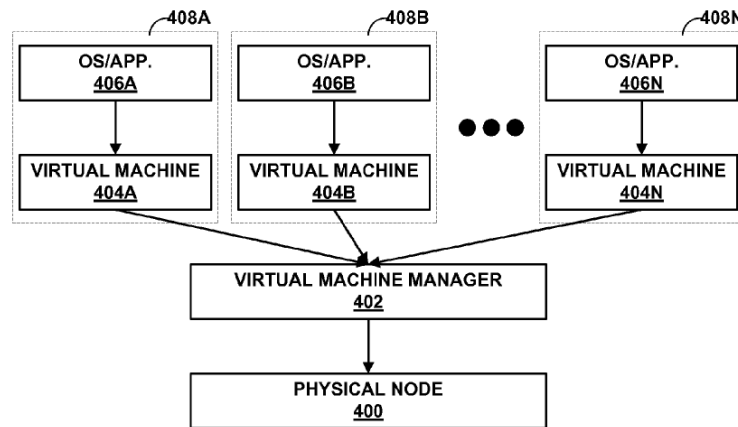


Figure 24 depicts a block diagram of a physical node to which a virtual machine manager and one or more virtual machines are deployed. *See* Ex. 1001, 3:28–30, 32:54–55. Physical node 400 may be any node within distributed computing system 10, for example, any one of application nodes 14. *See id.* at 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. Virtual machines 404 provide environments on which guest operating systems execute as though they were operating directly on a physical computing platform. *See id.* at 33:2–6. 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 27A of the '138 Patent is reproduced below.

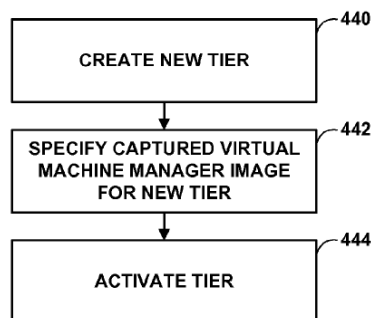


Figure 27A depicts a flow chart illustrating a procedure for autonomic deployment of a virtual machine manager to a node within the distributed computing system. *See Ex. 1001, 3:38–40, 35:25–27.* An administrator will create a new tier in step 440. *See id.* at 35:27–28. The administrator then specifies one or more captured virtual machine manager image instances for

assignment to the slots of the new tier in step 442. *See id.* at 35:27–28.

After the 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. *See id.* at 35:33–36. Control node 12 may then discover each of these virtual machines and place them in free pool 13 as if they were independent nodes. *See id.* at 35:36–38.

Figure 27B of the '138 Patent is reproduced below.

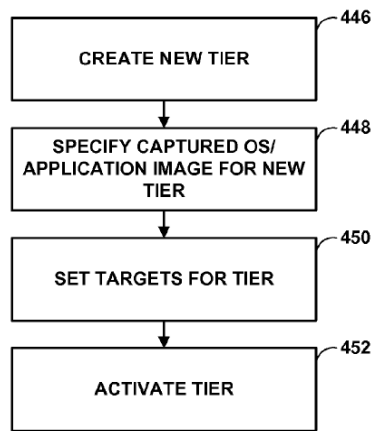


Figure 27B depicts a flow chart illustrating a procedure for deploying an operating system and application image to a node that is executing a virtual machine manager within the distributed computing system. *See Ex. 1001, 3:41–44, 35:48–51.* An administrator instructs the control node to create a new tier or select an existing tier in step 446. *See id.* at 35:51–53. The administrator then specifies a captured operating system and application image for the tier in step 448. *See id.* at 35:54–55. The administrator also sets target values for the image instances to be deployed to the slots of the new tier in step 450. *See id.* at 35:55–58. The administrator then activates the tier in step 452. *See id.* at 35:58. Next, based on the current state and goals of distributed computing system 10, control node 12 automatically selects a virtual machine from free pool 13, associates the virtual machine

with a slot in the tier, identifies the virtual disk image file (i.e., 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, and boots the virtual machine from the virtual disk image file. *See id.* at 35:59–67.

D. Illustrative Claim

Claims 1, 19, and 26 are independent, claims 2–5, 9, 10, and 17 depend from claim 1, and claims 20 and 21 depend from claim 19. Claim 1 is illustrative and reproduced below:

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.

E. Asserted Challenges to Patentability and Asserted Prior Art

Petitioner asserts that claims 1–5, 9, 10, 17, 19–21, and 26 would have been unpatentable on the following grounds:

Claim(s) Challenged	35 U.S.C. §	Reference(s)
1–5, 9, 19–21, 26	103 ¹	Khandekar ² ,
1–5, 9, 19–21, 26	103	Khandekar, Le ³
9, 10	103	Khandekar, Le, Sidebottom ⁴
17	103	Khandekar, Le, Stone ⁵

II. ANALYSIS

*A. Patent Owner’s Argument that Petitioner Did Not Provide
Governing Law, Rules, and Precedent*

Patent Owner contends that the Petition should be denied because it does not sufficiently include governing law, rules, or precedent as required by 37 C.F.R. § 42.22(a). *See* Prelim. Resp. 9. Patent Owner takes issue with the testimony of Petitioner’s declarant Dr. Shenoy regarding his understanding of patent law, arguing that Dr. Shenoy does not identify the specific governing law, rules, and precedent supporting his opinions and noting that Dr. Shenoy is not a lawyer or agent admitted to practice before the USPTO or Board. *See id.* (citing Ex. 1003, App. 4 at 212–224).

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. 1006, US 7,962,633 B1, issued June 14, 2011 (“Sidebottom”).

⁵ Ex. 1007, US 2003/0036886 A1, published Feb. 20, 2003 (“Stone”).

We disagree with Patent Owner. As a general matter, we find the Petition sufficiently explains the basis for Petitioner’s challenges to the claims of the ’138 Patent with citations to 35 U.S.C. § 103. *See* Pet. 9. Accordingly, we decline to deny the Petition on the basis of 37 C.F.R. § 42.22(a).

B. Claim Construction

For petitions filed after November 13, 2018, the Board applies the same claim construction standard as applied in federal courts in a civil action under 35 U.S.C. § 282(b), which is generally referred to as the *Phillips* standard. *See* 37 C.F.R. § 42.100(b) (2019); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). Under the *Phillips* standard, words of a claim are generally given their ordinary and customary meaning. *Phillips*, 415 F.3d at 1312.

Petitioner asserts that a person of ordinary skill in the art would have understood the phrase “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, to have been satisfied by, but not limited to, a plurality of virtual disk image files. *See* Pet. 13. Patent Owner does not agree with Petitioner’s construction. *See* Prelim. Resp. 5. Patent Owner does not offer a proposed construction, but instead asserts the phrase should be construed in accordance with its plain and ordinary meaning. *See id.* As demonstrated in the analysis below, for the purpose of institution, we need not expressly construe this claim phrase. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

Patent Owner’s arguments disputing Petitioner’s obviousness showing involve an issue of claim interpretation regarding claims 1, 19, and 26. *See*

Prelim. Resp. 10–16, 27–30. We address this issue in our analysis in Sections II.E.1.b.1.(f) and II.E.1.b.2 below.

C. 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.*, 383 U.S. 1, 17–18 (1966).⁶

D. 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. 12 (citing Ex. 1003 ¶¶ 33–34). Petitioner further contends that “[a] person with less formal education but more relevant practical experience may also meet” the

⁶ The parties do not present evidence of secondary considerations.

standard of a person of ordinary skill in the art. *Id.* Petitioner also asserts 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)⁷], 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. 12–13 (citing Ex. 1003 ¶ 35); *see* Ex. 1003 ¶ 12.

Patent Owner does not dispute Petitioner’s assertions addressing the skill level of a person of ordinary skill in the art. *See generally* Prelim.

Resp. For the purpose of institution, we adopt Petitioner’s definition of a person of ordinary skill in the art because it appears to be consistent with the

⁷ The parties use 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. *See, e.g.*, Pet. 6; Ex. 1003 ¶ 12; Prelim. Resp. 16–30; Ex. 1004, 4:67–5:1; Ex. 1005, 12:38. Patent Owner also uses the acronym VMM to refer to the claimed virtual machine manager. *See, e.g.*, Prelim. Resp. 3, 10–16. To eliminate 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.

level of skill reflected by the '138 Patent Specification and the asserted prior art.

E. Challenges to Patentability

1. Challenge to Claims 1–5, 9, 19–21, and 26 over Khandekar

a. Overview of Khandekar (Ex. 1004)

Khandekar is directed to the application of virtual machine 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.

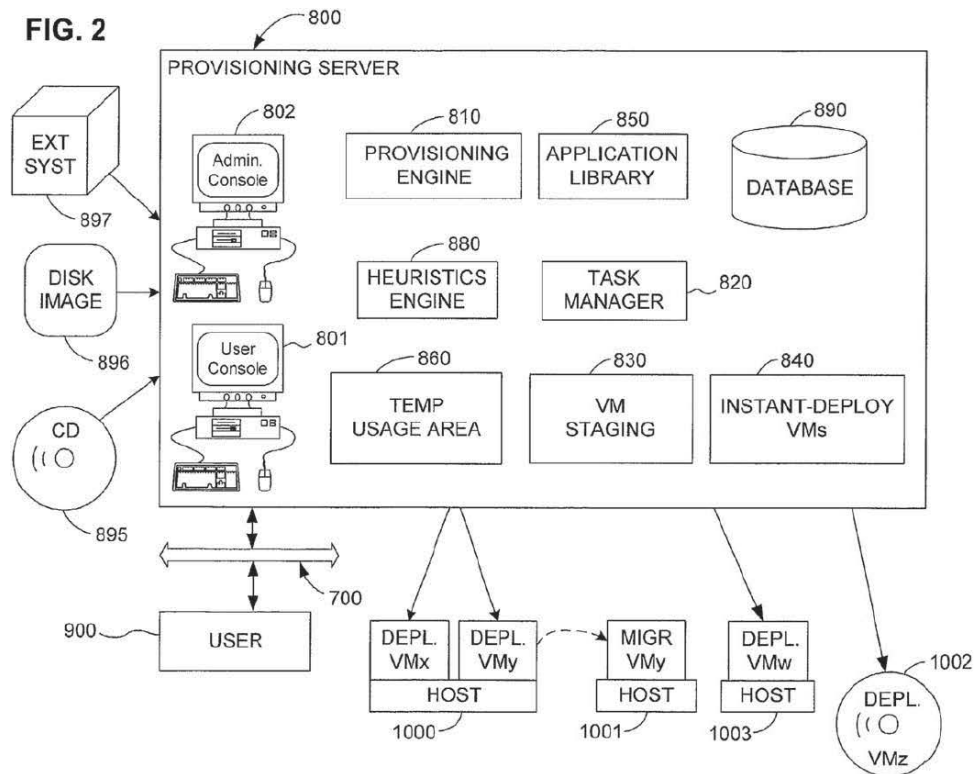


Figure 2 illustrates the main components of virtual machine (VM) provisioning. *See* Ex. 1004, 4:17–18, 9:10–12. Provisioning server 800 includes system hardware, system software, and devices as needed. *See id.* at 9:26–32. User console 801 allows end users to interact with provisioning server 800 via 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 operations 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 staged 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.

b. Analysis

For the reasons that follow, we are persuaded that Petitioner establishes sufficiently, for the purpose of institution, that Khandekar teaches, suggests, or renders obvious all of the limitations recited in independent claims 1, 19, and 26 and dependent claims 2–5, 9, 20, and 21.

1. Claim 1

- (a) 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

Petitioner contends that Khandekar teaches “[a] distributed computing system comprising: 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 network of cooperating VMs deployed on respective physical host platforms, and a provisioning server that includes various libraries for storing software images. *See* Pet. 21–23 (reproducing Ex. 1004, Fig. 2;

quoting Ex. 1004, 4:36–38, 5:9–11, 11:37, 12:40–44; citing Ex. 1004, code (57), 4:30–35, 4:38–41, 4:58–5:13, 9:26–32, 11:36–49, 12:44–50, 23:1–20, 23:26–51, 23:54–24:7, Fig. 7; Ex. 1003 ¶¶ 66–69, 74–77).⁸

Petitioner asserts that Khandekar discloses 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 23 (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, which obviously would have included hard disks.” *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. 26–27 (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 that Khandekar teaches 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 27 (quoting Ex. 1004, 8:41–44, 13:17–28; citing Ex. 1004, 8:36–41, 13:28–35; Ex. 1003 ¶ 82; incorporating by reference Pet. 41–47). Petitioner asserts a person of ordinary skill would have found it obvious to store VMM

⁸ Neither party asserts the preamble “[a] distributed computing system” is a claim limitation. *See generally* Pet.; Prelim. Resp. At this stage of the proceeding, we need not address or resolve whether the preamble is a claim limitation because Petitioner shows sufficiently that Khandekar teaches a distributed computing system.

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 23 (citing, e.g., Ex. 1004, 8:36–44, 13:17–35; Ex. 1003 ¶ 75).

For the purpose of institution and based on Petitioner’s citations to Khandekar and Dr. Shenoy’s testimony (Ex. 1003), we are persuaded Petitioner sets forth sufficient articulated reasoning with rational underpinning to support the conclusion 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, in addition to storing the VM images, in order to deploy the VMMs and VMs together. *See* Pet. 23, 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. 27 (citing Ex. 1004, 8:36–44, 13:17–35); *see id.* at 23 (citing Ex. 1004, 8:36–44, 13:17–35; Ex. 1003 ¶ 75; incorporating by reference Pet. 26–32), 28–29 (citing Ex. 1004, 4:36–41, 11:21–25, 16:43–63, 8:36–44, 13:17–35; 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 virtual machines, facilitating Khandekar’s goals of automated, flexible deployment of VMs to hosts.” *Id.* at 27–28 (citing Ex. 1004, 4:30–35; Ex. 1003 ¶ 83). Petitioner asserts that this modification would also remove the need to manually install VMMs on hosts, which is advantageous for a large number of hosts. *See id.* at 28 (citing Ex. 1004, 4:30–35; Ex. 1003 ¶ 83).

Patent Owner argues that Khandekar does not suggest “a software image repository comprising . . . a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes” because Khandekar states that the preferred method is to have the VMM pre-installed on each host. *See* Prelim. Resp. 22 (citing Ex. 1004, 13:17–35).

Patent Owner contends that Petitioner’s reliance on

[t]he isolated, single sentence disclosure in Khandekar, that “[i]t would be possible, however, for a corresponding VMM to be included with each staged VM, as long as the characteristics of the host on which the VM is be installed are known,” provides no teaching or suggestion about how a VMM might be stored, let alone teaching “VMM images being stored in the provisioning server.”

Prelim. Resp. 24 (quoting Ex. 1004, 13:25–28; citing Ex. 1003 ¶ 83); *see id.* at 22–23 (quoting Pet. 27). Patent Owner also contends that “the proposition espoused by Petitioner’s expert, that ‘A [person of ordinary skill in the art] 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.” *Id.* at 23–24 (quoting Ex. 1004, 13:25–28; Ex. 1003 ¶ 83).

On the current record, we do not agree with Patent Owner’s arguments. Patent Owner’s arguments are premised on a requirement for an explicit teaching in Khandekar of storing a plurality of VMMs. It is not necessary, however, to find precise teachings in the prior art directed to the specific subject matter claimed because inferences and creative steps that a person of ordinary skill in the art would employ can be taken into account. *See KSR*, 550 U.S. at 418. Here, Petitioner relies on the creative step that

one with ordinary skill in the art would have employed—to store a plurality of VMM images in the provisioning server in order to deploy VMs and corresponding VMMs together to hosts—because Khandekar teaches storing VM images in a provisioning server and teaches including and installing a corresponding VMM with each staged VM as long as the host characteristics are known. *See* Pet. 23, 26–29.

Patent Owner also argues that Petitioner’s analysis is pure hindsight, which is forbidden in an obviousness analysis. *See* Prelim. Resp. 24–25 (citations omitted). To this end, Patent Owner contends that Khandekar teaches away from the stated possibility for the “VMM to be included along with each staged VM” because in the next sentence, Khandekar teaches that “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.” *See id.* at 25 (quoting Ex. 1004, 13:25–32). Patent Owner asserts that this teaching away from an alternative embodiment negates any argument that a person of ordinary skill in the art would find reason to modify Khandekar. *See id.*

We do not agree, on the current record, that Khandekar teaches away from Petitioner’s proposed modification. “Under the proper legal standard, a reference will teach away when it suggests that the developments flowing from its disclosures are unlikely to produce the objective of the applicant’s invention.” *Syntex (U.S.A.) LLC v. Apotex, Inc.*, 407 F.3d 1371, 1380 (Fed. Cir. 2005) (citing *In re Gurley*, 27 F.3d 551, 553 (Fed. Cir. 1994)). “A reference does not teach away, however, if it . . . does not ‘criticize, discredit, or otherwise discourage’ investigation into the invention claimed.” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 567 F.3d 1314, 1327

(Fed. Cir. 2009) (quoting *In re Fulton*, 391 F.3d 1195, 1201 (Fed. Cir. 2004)). At this stage of the proceeding, we do not view Khandekar's teaching that it is advantageous to pre-install a VMM, without a VM, on a plurality of hosts to criticize, discredit, or otherwise discourage investigation into storing a plurality of VMMs on the provisioning server and including a corresponding VMM along with each VM to be installed on a host.

In sum, based on Petitioner's persuasive showing, summarized above, we determine Petitioner shows sufficiently for institution that Khandekar teaches, suggests, or renders obvious these recitations of claim 1.

(b) 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. Pet. 37–38 (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 39–40 (quoting Ex. 1004, 6:23–43, 7:45–47; citing Ex. 1004, 6:44–61, 7:39–44; Ex. 1003 ¶ 115).

Patent Owner does not individually dispute these assertions. *See* Prelim. Resp. 21–29. Nonetheless, the burden remains on Petitioner to

demonstrate unpatentability. *See Dynamic Drinkware, LLC v. National Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015). At this stage of the proceeding, we determine Petitioner shows sufficiently that Khandekar teaches this limitation of claim 1.

(c) (ii) a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines; and

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,” as recited in claim 1, based on Khandekar’s disclosure of a plurality of virtual disk image files comprising pre-built, custom-built, and instantly-deployable VMs. *See* Pet. 41–42 (quoting Ex. 1004, 9:10–25; citing Ex. 1004, 10:15–26, 14:4–8, 18:56–19:2; Ex. 1003 ¶¶ 122–139, Pet. 13–16, 55–60); *see also id.* at 42–47 (explaining in more detail the virtual disk images for pre-built, custom-built, and instantly-deployable VMs) (citations omitted). Petitioner further explains that each virtual disk image includes an operating system and application data. *See id.* at 42 (citing Ex. 1003 ¶ 123), 47 (citing Ex. 1004, 6:43–56; Ex. 1003 ¶ 137). According to Petitioner, “Khandekar teaches custom-built VMs where application images (installation files) are stored in an application library and copied to hosts.” *Id.* at 42 (citing Ex. 1003 ¶ 124); *see id.* at 44–46 (citations omitted).

Patent Owner does not individually dispute Petitioner’s assertions based on Khandekar addressing this limitation of claim 1. *See* Prelim. Resp. 21–29. In any event, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378. At this stage of

the proceeding, we determine Petitioner shows sufficiently that Khandekar teaches this limitation of claim 1.

(d) 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

Petitioner asserts that Khandekar teaches “a control node that comprises an automation infrastructure to provide autonomic deployment,” as recited in claim 1, based on Khandekar’s disclosure of provisioning server 800 that comprises a provisioning engine, a heuristics/rules engine, and supporting functionality, and disclosure that the invention enables users to configure, deploy, and get access to substantially customized computer systems with no need for human intervention. *See* Pet. 47–48 (reproducing Ex. 1004, Fig. 2; quoting Ex. 1004, 5:25–28; citing Ex. 1004, 5:28–36, 9:9–25, 23:54–24:27; Ex. 1003 ¶¶ 140–147). Petitioner contends that Khandekar teaches selecting one of the preconfigured VMs, which is then automatically prepared for deployment, and that, after a request is made, heuristics engine 880 applies heuristics/rules to select which host machine is the most appropriate on which to deploy the VM. *See id.* at 49 (quoting Ex. 1004, 4:39–41, 4:58–63, 10:29–33; citing Ex. 1004, 4:36–39, 4:63–65, 5:25–36, 9:9–25, 10:27–29, 10:33–42, 23:28–51, Figs. 2, 7; Ex. 1003 ¶¶ 141–142; Pet. 22–47, 55–60). Petitioner asserts a person of ordinary skill in the art would have found it obvious to autonomically deploy image instances of VMMs from the provisioning server to the hosts because Khandekar teaches automatic deployment of VM images and providing a

VMM together with a VM. *See id.* at 49–50 (citing, e.g., Ex. 1004, 4:36–41, 4:58–65, 10:27–42, 8:36–44, 13:17–35; Ex. 1003 ¶¶ 143–144).

For the purpose of institution and based on Petitioner’s citations to Khandekar and Dr. Shenoy’s testimony (Ex. 1003), we are persuaded Petitioner sets forth sufficient articulated reasoning with rational underpinning to support the conclusion that it would have been obvious to one of ordinary skill in the art to modify Khandekar’s teachings to include autonomically deploying image instances of VMMs from the provisioning server to the hosts. *See* Pet. 47–50; *KSR*, 550 U.S. at 418. Specifically, Petitioner contends that Khandekar teaches autonomic deployment of images is accomplished by copying the image from the provisioning server’s repository to the host computer without human intervention. *See* Pet. 50 (citing Ex. 1004, 4:36–41, 5:25–36, 18:56–19:2). Petitioner asserts Khandekar teaches that the autonomic deployment includes copying VM images and that a VMM “is included and installed together with the VM.” *See id.* (quoting Ex. 1004, 8:43–44; citing Ex. 1004, 8:36–43, 13:25–28; incorporating by reference Pet. 55–60). According to Petitioner, “[s]ince the VMM is included with the VM, it obviously is copied along with the VM to the host computer.” *Id.* (citing Ex. 1003 ¶ 145). Petitioner asserts that this conventional usage of VMM images was known and used in the art. *See id.* (citing Ex. 1014 ¶ 62; Ex. 1016 ¶¶ 43, 49; Ex. 1003 ¶ 145). Petitioner asserts that a person of ordinary skill in the art would have found it obvious to autonomically deploy image instances of VMMs from the provisioning server to the hosts because Khandekar teaches automatic deployment of VM images and teaches providing a VMM together with the VM. *See id.* at 49–50 (quoting Ex. 1004, 8:41–44; citing Ex. 1004, 4:36–41, 4:58–65,

8:36–41, 10:27–42, 13:17–35; Ex. 1003 ¶¶ 143–144; incorporating by reference Pet. 13–20, 26–32).

Patent Owner does not individually dispute Petitioner’s assertions based on Khandekar addressing this limitation of claim 1. *See* Prelim. Resp. 21–29. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378. At this stage of the proceeding, we determine Petitioner shows sufficiently that Khandekar, modified as discussed above, teaches this limitation of claim 1.

(e) 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.

Petitioner asserts that Khandekar teaches the control node comprises an automation infrastructure “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,” as recited in claim 1, based on Khandekar’s disclosure that the provisioning server controls the automated deployment of VM images from the provisioning server’s repository to host computers, which includes installing additional applications in the deployed VM if not pre-installed and requested in the user information. *See* Pet. 55–57 (reproducing Ex. 1004, Fig. 7 (with emphasis added); citing Ex. 1004, 4:36–41, 4:58–65, 5:25–36, 10:27–42, 23:28–51, 28:54–29:27 (claim 1); Ex. 1003 ¶¶ 155–163; incorporating by reference Pet. 41–55); *see also id.* at 57–60 (explaining in more detail the deployment of virtual disk images for pre-built, custom-built, and instantly-deployable VMs) (citations omitted).

Patent Owner does not individually dispute Petitioner's assertions based on Khandekar addressing this limitation of claim 1. *See* Prelim. Resp. 21–29. In any event, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378. At this stage of the proceeding, we determine Petitioner shows sufficiently that Khandekar teaches this limitation of claim 1.

(f) Patent Owner's Additional Arguments

Patent Owner argues claim 1 covers the process of installing a VM and the additional application software, including the following “three steps: (1) copying a [virtual machine manager] image onto a target node; (2) executing the [virtual machine manager] image to create a working VM environment; and (3) copying a software application image into the VM environment.” *See* Prelim. Resp. 10; *see also id.* at 11–12 (reproducing claim 1 with color-coded text for certain recitations). According to Patent Owner,

Claim 1 requires the control node to perform three fundamental operations . . . :

1. causing the plurality of image instances of the virtual machine manager to be copied from the software image repository to the application nodes;
2. execut[ing], on the application nodes, the image instances of the virtual machine manager [to] provide a plurality of virtual machines; and
3. provid[ing] autonomic deployment of the plurality of image instances of the software application on the virtual machines by causing the plurality of image instances of the software application to be copied from the software image repository to the application nodes.

Id. at 12–13 (alterations in original). According to Patent Owner, the “[third] step requires the image instances of the software application to be deployed on the VMs. The VMs themselves are only created (provided) when the [virtual machine manager] image instances are executed.” *Id.* at 13. Patent Owner reasons that the virtual machine manager image instances must be executed in the second step before the image instances of the software application can be deployed. *See id.*

Patent Owner contends that Khandekar fails to teach or suggest this three-step sequence that claim 1 requires. *See* Prelim. Resp. 27–28. Patent Owner asserts that neither of Khandekar’s methods of deploying the VMM, i.e., pre-installing the VMM or installing the VMM with the VM, discloses or suggests that a VMM image instance might be deployed and executed before the deployment of the VM itself. *See id.* at 28. Patent Owner further contends that Khandekar’s statement regarding a VMM being included along with each staged VM also does not disclose this sequence because Khandekar’s disclosure: (1) requires that the VMM is installed with the software image instance, not prior to the software application image instance; (2) provides no suggestion that the VMM image instance is first executed to create the VM prior to the installation of the software applications image instances; and (3) fails to suggest that the software application image instances are deployed onto the VM provided by the VMM image instance. *See id.* at 28–29.

On the current record, we are not persuaded by Patent Owner’s arguments because claim 1 does not positively recite an installation process that requires the performance of these three steps in an ordered sequence, as argued by Patent Owner. *See* Ex. 1001, 36:26–49. Claim 1 also does not

recite or require: (1) executing the virtual machine manager image to *create* a working VM environment and copying a software application image into the VM environment; (2) deployment and execution of the virtual machine manager before the deployment of the VM; or (3) installation of the software applications instances. *Compare* Prelim. Resp. 10, 12–13, *with* Ex. 1001, 36:27–35, 36:39–49.⁹

(g) Summary

For all of the foregoing reasons, for the purpose of institution and based on the record before us, Petitioner establishes sufficiently that the subject matter of claim 1 would have been obvious over Khandekar.

2. Independent Claims 19 and 26

Independent claim 19 recites a method that includes limitations similar to those recited in claim 1. *Compare* Ex. 1001, 38:1–21, *with id.* at 36:26–49. In addition to relying on its arguments addressing the limitations of claim 1, Petitioner asserts that Khandekar teaches or renders obvious “[a] method,” as recited in claim 19, based on Khandekar’s disclosure of a method and teachings of various steps and algorithms performed by the provisioning server. *See* Pet. 66 (quoting Ex. 1004, 4:30–31; citing Ex. 1004, 4:31–35, 23:1–20, 23:26–51, Fig. 7; Ex. 1003 ¶¶ 194–197; incorporating by reference Pet. 21–60).¹⁰ Petitioner also

⁹ The 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 Patent Owner.

¹⁰ Neither party asserts the preamble “[a] method” is a claim limitation. *See generally* Pet.; Prelim. Resp. At this stage of the proceeding, we need not address or resolve whether the preamble is a claim limitation because Petitioner shows sufficiently that Khandekar teaches a method.

contends that Khandekar teaches or renders obvious claim 19's step of "storing a plurality of image instances . . . in a distributed computing system." *See id.* at 67–68 (citing Ex. 1003 ¶¶ 200–206; incorporating by reference Pet. 21–47). Petitioner further asserts that Khandekar teaches the recited step of "executing a rules engine to perform operations to autonomically deploy the image instances," based on Khandekar's disclosure of a heuristics/rules engine. *See id.* at 68–70 (citing Ex. 1003 ¶¶ 207–214; incorporating by reference Pet. 47–60).

Claim 26 recites similar limitations to those recited in independent claims 1 and 19, in the form of a non-transitory, computer-readable medium. *Compare* Ex. 1001, 38:61–39:15 (claim 26), *with id.* at 36:26–49 (claim 1), *and id.* at 38:1–21 (claim 19). Petitioner asserts that Khandekar teaches, suggests, or renders obvious the limitations of claim 26, relying primarily on its arguments addressing claim 19. *See* Pet. 71–74 (citing Ex. 1003 ¶¶ 222–224, 228–232; incorporating by reference Pet. 67–70). Petitioner contends that Khandekar teaches or renders obvious "[a] non-transitory, computer readable medium comprising instructions stored thereon," as recited in claim 26, based on Khandekar's disclosure that the provisioning server includes standard computer components that would have included a processor to execute software instructions and used a hard disk to store the software.¹¹ *See id.* at 71–72 (citing Ex. 1004, 9:26–45; Ex. 1003 ¶¶ 222–224).

¹¹ Neither party asserts the preamble "[a] non-transitory, computer-readable medium comprising instructions stored-thereon, that when executed by a programmable processor" is a claim limitation. *See generally* Pet.; Prelim. Resp. At this stage of the proceeding, we need not address or resolve

Patent Owner argues that claims 19 and 26, as with claim 1, require the performance of a three-step ordered sequence, specifically “that [virtual machine manager] image instances be deployed, and then that software application image instances be deployed onto VMs created by the [virtual machine manager] image instances.” Prelim. Resp. 15; *see id.* at 13–15 (reproducing claims 19 and 26 with color-coded text for certain recitations). Patent Owner asserts “the [virtual machine manager] image instance must be deployed and executed before the software application image instance can be deployed onto the VM” because “the [virtual machine manager] image instances provide the VMs by executing on the application nodes.” *Id.* at 15–16. On the current record, for the same reasons as those explained above addressing Patent Owner’s arguments regarding claim 1, we are not persuaded by Patent Owner’s arguments.¹²

For all of these reasons, in addition to those explained above addressing claim 1, for the purpose of institution and based on the record before us, Petitioner establishes sufficiently that the subject matter of independent claims 19 and 26 would have been obvious over Khandekar.

3. Dependent Claims 2–5, 9, 20, and 21

Claims 2–5 and 9 depend from claim 1 and claims 20 and 21 depend from independent claim 19. Ex. 1001, 36:50–63, 37:9–13, 38:22–27. We have reviewed Petitioner’s contentions and supporting evidence addressing

whether the preamble is a claim limitation because Petitioner shows sufficiently that Khandekar teaches the recitation in the preamble.

¹² As with claim 1, the parties are 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.

how Khandekar teaches, suggests, or renders obvious the additional limitations recited in dependent claims 2–5, 9, 20, and 21. *See* Pet. 60–66, 71 (citations omitted). Patent Owner does not address substantively Petitioner’s contentions based on Khandekar addressing the additional limitations of dependent claims 2–5, 9, 20, and 21. *See* Prelim. Resp. 16–29. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378.

For the purpose of institution and based on the record before us, Petitioner establishes sufficiently that the subject matter of dependent claims 2–5, 9, 20, and 21 would have been obvious over Khandekar.

c. Summary

For all of the foregoing reasons, for the purpose of institution and based on the record before us, we determine there is a reasonable likelihood Petitioner would prevail in showing that claims 1–5, 9, 19–21, and 26 are unpatentable under 35 U.S.C. § 103 over Khandekar.

2. Challenge to Claims 1–5, 9, 19–21, and 26 over Khandekar and Le

a. 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, 60:64–7:3, 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.

b. Analysis

For the reasons that follow, we are persuaded that Petitioner establishes sufficiently, for the purpose of institution, that the combination of Khandekar and Le teaches, suggests, or renders obvious all of the

limitations recited in independent claims 1, 19, and 26 and dependent claims 2–5, 9, 20, and 21.

1. Claim 1

- (a) 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

Petitioner asserts that, in addition to Khandekar’s teachings discussed above in Section II.E.1.b.1.(a), 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 manages a set of physical computers and virtual machines residing on a network. *See* Pet. 22 (quoting Ex. 1005, 59:29–30; citing Ex. 1005, 29:30–36, 53:45–57, 60:19–28; Ex. 1003 ¶¶ 70–73; incorporating by reference Pet. 16–20). Petitioner contends that 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 id.* at 24–25 (reproducing Ex. 1005, Fig. 6; 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; incorporating by reference Pet. 16–20, 32–37, 40–41). 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 virtual machine and the host’s physical resources, and disclosure of the use of VMware ESX software. *See id.* at 32–33 (reproducing Ex. 1005, Fig. 8; quoting Ex. 1005,

65:65–66:2, 66:11–13; citing Ex. 1005, 13:22–30, 32:62–64, 66:13–32, 67:20–32, 66:48–64, 67:9–19; Ex. 1003 ¶¶ 96–98; incorporating by reference Pet. 16–20).

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. 34 (citing Ex. 1003 ¶¶ 99–111; incorporating by reference Pet. 16–21). Specifically, Petitioner contends that Le teaches implementation details of image creation and management for distributed computing and virtual machine 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 17 (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 (which would include a Type 1 VMM) and applications (which would include a Type 2 VMM).” *Id.* at 18 (citing Ex. 1005, 1:64–3:13, 2:19–21, 15:63–65, 16:29–30, 54:14–21; incorporating by reference Pet. 26–36, 47–55); *see also id.* at 34 (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 35–36 (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; incorporating by reference Pet. 16–20).

Petitioner contends Le teaches that its system determines the hardware characteristics of the remote computer, and analyzes the computer's hardware. *See id.* at 19 (quoting Ex. 1005, 70:21–24, citing Ex. 1005, 70:13–35; Ex. 1003 ¶ 59). Petitioner also asserts that Le explains it was common to create one image file per family of similar computers. *See id.* at 36 (citing Ex. 1005 9:66–10:9, 16:29–41; Ex. 1003 ¶ 106).

For the purpose of institution and based on Petitioner's citations to Khandekar, Le, and Dr. Shenoy's testimony (Ex. 1003), we are persuaded Petitioner sets forth sufficient articulated reasoning with rational underpinning to support the conclusion 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. 17 (citing Ex. 1003 ¶ 59), 34–35 (citing Ex. 1003 ¶ 102), 36 (citing Ex. 1003 ¶ 105).

Patent Owner does not individually dispute Petitioner's assertions based on the combined teachings of Khandekar and Le addressing these limitations of claim 1. *See* Prelim. Resp. 29–30. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378. At this stage of the proceeding, we determine Petitioner shows sufficiently that the Khandekar-Le combination teaches these recitations of claim 1.

(b) 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, in addition to Khandekar's teachings addressed above in Section II.E.1.b.1.(b), Le teaches "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," as recited in claim 1, based on Le's disclosure of a VMM and a virtual machine manager that support a plurality of virtual machines on a host. *See* Pet. 40–41 (reproducing Ex. 1005, Fig. 8; quoting Ex. 1005, 65:43–50; citing Ex. 1005, 65:65–66:32; Ex. 1011, 1, 5, Fig. 1; Ex. 1016 ¶ 76; Ex. 1003 ¶¶ 117–121; incorporating by reference Pet. 16–20, 26–37).

Patent Owner does not individually dispute Petitioner's assertions based on the combined teachings of Khandekar and Le addressing this limitation of claim 1. *See* Prelim. Resp. 29–30. In any event, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378. At this stage of the proceeding, we determine Petitioner shows sufficiently that the Khandekar-Le combination teaches this limitation of claim 1.

(c) (ii) a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines; and

As explained above in Section II.E.1.b.1.(c), 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," as recited in claim 1. *See*

Pet. 41–42 (citations omitted). Petitioner does not rely on Le for teaching this limitation. *See id.* at 41–47.

Patent Owner does not individually dispute Petitioner’s assertions based on Khandekar addressing this limitation of claim 1. *See* Prelim. Resp. 29–30. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378. As explained above in Section II.E.1.b.1.(c), we determine Petitioner shows sufficiently for the purpose of institution that Khandekar teaches this limitation of claim 1.

(d) 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

For the limitation of claim 1 that recites:

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,

Petitioner asserts that Le’s teachings “complement and elaborate on” Khandekar’s disclosures that teach or suggest this limitation, which are addressed above in Section II.E.1.b.1.(d). Pet. 51 (citing Ex. 1003 ¶¶ 148–154). In particular, Petitioner relies on Le’s disclosures of: (1) disk imaging techniques for Type 1 and Type 2 VMMs; (2) conventional use of disk imaging to clone an image of a computer, including the operating system and applications onto other machines; (3) techniques for copying images from a central repository to remote destination computers; and

(4) the UCMS deploying images to virtual machines and physical computers, including guiding deployment by analyzing the hardware configuration of new computers. *See id.* at 51–54 (reproducing Ex. 1005, Figs. 3–5 (emphasis added); quoting Ex. 1005, 10:55–62, 27:5–6, 72:21–22; citing Ex. 1005, 2:19–21, 8:10:23, 20:25–37, 22:32–44, 26:52–27:9, 54:14–21, 59:29–36, 65:20–33, 70:12–35; 72:14–21, Fig. 9; Ex. 1003 ¶¶ 148–154; incorporating by reference Pet. 16–20, 26–37, 55–60).

Petitioner contends that a person of ordinary skill in the art would have been motivated to use Le’s UCMS functionality of analyzing the hardware configuration of new computers to address Khandekar’s teaching that host characteristics should be determined before sending a VMM. *See id.* at 54–55 (citing Ex. 1004, 13:17–15; Ex. 1005, 70:12–35); *see also id.* at 19 (similar argument; quoting Ex. 1005, 70:21–24; citing Ex. 1005, 70:13–35; Ex. 1003 ¶ 59). According to Petitioner, “it would have been obvious for the provisioning server to deploy VMM images to physical computers and VM images for virtual machines, using the hardware configuration of the host computer to guide the automation process and send an appropriate image to the host computer.” *Id.* at 55 (citing Ex. 1005, 59:29–36, 65:20–33, 70:12–35; Ex. 1003 ¶¶ 152–153).

Patent Owner does not individually dispute Petitioner’s assertions based on the combined teachings of Khandekar and Le addressing this limitation of claim 1. *See* Prelim. Resp. 29–30. In any event, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378.

For all of the foregoing reasons, we are persuaded Petitioner shows sufficiently that the Khandekar-Le combination teaches this limitation of

claim 1. Further, Petitioner has set forth sufficient articulated reasoning with rational underpinning to support the conclusion that it would have been obvious to one of ordinary skill in the art to modify Khandekar's provisioning server, based on Le's teachings, to analyze the hardware configuration of the host computer, in order to guide the automation process and send the appropriate software application image to the host computer.

See Pet. 47–54; *KSR*, 550 U.S. at 418.

(e) 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.

As explained above in Section II.E.1.b.1.(e), 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,” as recited in claim 1. *See* Pet. 55–60 (citations omitted). Petitioner does not rely on Le for this limitation. *See id.*

Patent Owner does not individually dispute Petitioner's assertions based on Khandekar addressing this limitation of claim 1. *See* Prelim. Resp. 29–30. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378. As explained in Section II.E.1.b.1.(e), we determine Petitioner shows sufficiently for the purpose of institution that Khandekar teaches this limitation of claim 1.

(f) Patent Owner's Arguments

Patent Owner argues that Le fails to remedy the deficiencies of Khandekar, in particular, that the Petition fails to identify any teaching or suggestion in Le of the three-step deployment process required by

independent claim 1. *See* Prelim. Resp. 29–30. For the same reasons as those explained in Section II.E.1.b.1.(f), on the current record, we do not agree with Patent Owner’s arguments.

(g) Summary

For all of the foregoing reasons, for the purpose of institution and based on the record before us, Petitioner establishes sufficiently that the subject matter of claim 1 would have been obvious over the combination of Khandekar and Le.

2. Independent Claims 19 and 26

For independent claim 19, Petitioner asserts that, in addition to Khandekar’s teachings addressed above in Section II.E.1.b.2, Le teaches a method, as recited in claim 19, based on Le’s disclosure of an image capture and deployment methods and various steps and algorithms. *See* Pet. 66 (citing Ex. 1005, 59:21–37, 64:59, 65:21–33; Ex. 1003 ¶¶ 198–199; incorporating by reference Pet. 21–60). Petitioner incorporates by reference arguments addressing claim 1 and further asserts that the combination of Khandekar and Le teaches or renders obvious “storing a plurality of image instances . . . in a distributed computing system.” *See id.* at 67–68 (citing Ex. 1003 ¶¶ 200–206; incorporating by reference Pet. 22–47).

For independent claim 26, Petitioner asserts that the combination of Khandekar and Le teaches, suggests, or renders obvious each limitation, relying primarily on its arguments addressing claim 19. *See* Pet. 71–74 (citing Ex. 1003 ¶¶ 222–224, 228–232; incorporating by reference Pet. 67–70). Petitioner contends that, in addition to Khandekar’s teachings addressed above in Section II.E.1.b.2, Le teaches “[a] non-transitory

computer readable medium comprising instructions stored thereon,” as recited in claim 26, based on Le’s disclosure of using software installed on a computer to implement the features of the UCMS and teachings of software programs comprising computer-executable instructions stored on disks. *See id.* at 72 (citing Ex. 1005, 23:44–52, 26:38–52, 59:29–36, 60:13–18, 89:8–31, Fig. 4; Ex. 1003 ¶¶ 225–227; incorporating by reference Pet. 12–16).

Patent Owner argues that Le fails to remedy the deficiencies of Khandekar with respect to the limitations of claims 19 and 26 for the same reasons as claim 1. *See* Prelim. Resp. 29–30. For the same reasons as those explained in Section II.E.1.b.1.(f), and on the current record, we do not agree with Patent Owner’s arguments.

For all of the foregoing reasons, in addition to those explained above addressing claim 1, for the purpose of institution and based on the record before us, Petitioner establishes sufficiently that the subject matter of claims 19 and 26 would have been obvious over the combination of Khandekar and Le.

3. Dependent Claims 2–5, 9, 20, and 21

We have reviewed Petitioner’s contentions and supporting evidence addressing how the combination of Khandekar and Le teaches, suggests, or renders obvious the additional limitations recited in dependent claims 2–5, 9, 20, and 21. *See* Pet. 60–66, 71 (citations omitted). Patent Owner does not address substantively Petitioner’s contentions based on the combination of Khandekar and Le addressing the additional limitations of dependent claims 2–5, 9, 20, and 21. *See* Prelim. Resp. 29–30. Nonetheless, the

burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378.

For the purpose of institution and based on the record before us, Petitioner establishes sufficiently that the subject matter of dependent claims 2–5, 9, 20, and 21 would have been obvious over the combination of Khandekar and Le.

c. Summary

For all of the foregoing reasons, for the purpose of institution and based on the record before us, we determine there is a reasonable likelihood Petitioner would prevail in showing that claims 1–5, 9, 19–21, and 26 are unpatentable under 35 U.S.C. § 103 over Khandekar and Le.

3. Challenge to Claims 9 and 10 over Khandekar, Le, and Sidebottom

a. Overview of Sidebottom (Ex. 1006)

Sidebottom is directed to managing computer networks by defining and deploying business policies using a Service Deployment Device (SDD). *See Ex. 1006*, 1:7–8, 1:36–44. A service provider, such as an Internet Service Provider, provides subscribers with access to the internet in accordance with a service level agreement (SLA). *See id.* at 2:29–30. The SLA is a contract constituting the business policies of the service provider to the subscriber, including the terms of service. *See id.* at 2:34–50.

Subscriber devices communicate with an edge router to forward network communications to service provider devices. *See id.* at 2:51–66. The edge router and service provider devices generate and send network-level events to the SDD. *See id.* at 2:67–3:17. The SDD processes the network events to deploy and enforce the service provider’s network policies by applying a set of business logic rules to the network events to perform actions related to the

services provided. *See id.* at 3:18–34. Business rules specify a set of condition/action rules embodying the business policies of the service provider. *See id.* at 3:34–40. When the SDD receives an event, the SDD evaluates the condition of the condition/action rule, and if the SDD determines the condition has been satisfied given the event, the SDD enables the action of the condition/rule. *See id.* at 3:40–43; *see also id.* at 9:36–57, Fig. 5 (describing an example where a change in working memory is the condition of a condition/action rule). Business rules may include conditions using arithmetic, Boolean, or other operators, for example, to combine business-level information. *See id.* at 3:51–56. “In one embodiment, [the] SDD . . . uses a forward-chaining rule-based algorithm.” *Id.* at 3:56–57; *see id.* at 9:43–45.

b. Analysis

For the reasons that follow, we are persuaded that Petitioner establishes sufficiently, for the purpose of institution, that the combination of Khandekar, Le, and Sidebottom teaches, suggests, or renders obvious all of the limitations recited in dependent claims 9 and 10.

Claim 9 depends from claim 1 and additionally recites “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.” Ex. 1001, 37:9–13. Petitioner contends that Khandekar teaches or suggests this additional limitation of claim 9. *See* Pet. 64 (citing Ex. 1004, 10:27–42), 74 (incorporating by reference Pet. 47–60, 64–66). In particular, Petitioner contends that Khandekar discloses the provisioning server includes a heuristics engine, which uses rules, may include other known knowledge-based systems to

direct deployment of a selected staged VM to the host processor that currently has the most available cycles, and monitors whether a host is overloaded. *See id.* at 64–66 (quoting Ex. 1004, 4:61–63, 10:31–42, 12:65–13:8, 13:51–52; citing Ex. 1004, 4:58–65, 10:27–31, 11:11–19, 12:24–39, 12:57–65, 13:44–56, 14:4–18, 14:35–49, 20:31–62, 23:54–24:27, Fig. 2; Ex. 1003 ¶¶ 187–193), 78 (citing Ex. 1004, 4:58–65, 13:44–56).

Petitioner also contends that Sidebottom teaches a rules engine that uses a forward-chaining rule-based algorithm. *See id.* at 77–78 (reproducing Ex. 1006, Fig. 2; quoting Ex. 1006, 1:41–44, 9:36–45; citing Ex. 1006, 2:65–3:17, 3:32–43, 4:26–44, 5:63–6:17, 3:44–58; Ex. 1003 ¶¶ 247–248).

Claim 10 depends from claim 9 and further recites “the one or more rule engines are forward-chaining rule engines.” Ex. 1001, 37:14–16. Petitioner acknowledges that the combination of Khandekar and Le does not teach or suggest a forward-chaining rule engine. *See* Pet. 74. Khandekar discloses “[k]nown expert systems, ‘intelligent assistants,’ and other knowledge-based systems and engines that can be embedded in other products and loaded with domain knowledge may be included in the heuristics engine 880 to implement the various decision functions.” Ex. 1004, 10:38–42. Petitioner contends that Sidebottom provides a teaching of a rules engine that applies a forward-chaining rule-based algorithm. *See* Pet. 74 (citing Ex. 1006, 3:44–58, 9:36–45), 79 (quoting Ex. 1006, 3:56–58, 9:43–45; citing Ex. 1006, 3:44–56, 9:36–43; Ex. 1003 ¶ 253).

For the purpose of institution and based on Petitioner’s citations to Khandekar, Le, Sidebottom, and Dr. Shenoy’s testimony (Ex. 1003), we determine Petitioner shows sufficiently that Khandekar alone and together

with Sidebottom teaches the additional limitation of claim 9, and Sidebottom teaches the additional limitation of claim 10. We are further persuaded Petitioner sets forth sufficient articulated reasoning with rational underpinning to support the conclusion that it would have been obvious to one of ordinary skill in the art to modify the combined teachings of Khandekar and Le to include a forward-chaining rule engine, as taught by Sidebottom. *See* Pet. 74–79; *KSR*, 550 U.S. at 418. Petitioner asserts that Khandekar provides motivation to apply known knowledge-based systems and engines to its heuristics/rules engine. *See* Pet. 74 (citing Ex. 1004, 10:27–42; Ex. 1003 ¶¶ 235–236). Petitioner further contends that Sidebottom provides motivation to combine its teachings with Khandekar because Sidebottom’s teachings are broadly applicable for defining policies in a computer network, and Sidebottom’s high-level rules were adaptable to a wide variety of applications and business needs. *See id.* at 75 (citing Ex. 1006, 1:36–38), 76 (citing Ex. 1003 ¶ 240). Petitioner asserts that both Sidebottom and Khandekar teach the use of rules for defining condition/action rules based on monitored events in a network. *See id.* at 76 (citing Ex. 1004, 10:27–42, 13:44–56; Ex. 1006, 3:32–43, 4:26–44, 5:63–6:17). Petitioner contends that forward-chaining rules engines could be readily utilized in the combination of Khandekar and Le because they are general algorithms, not tied to any particular implementation. *See id.* (citing Ex. 1003 ¶ 241). Petitioner asserts that the combination is the application of known techniques (e.g., forward chaining rules algorithms) to improve similar devices (e.g., distributed computing systems in a network) in the same way as they were used in the prior art. *See id.* at 77 (citing Ex. 1003 ¶ 243).

Patent Owner does not address substantively Petitioner's contentions based on the combination of Khandekar, Le, and Sidebottom addressing the additional limitations of claims 9 and 10. *See* Prelim. Resp. 31. In any event, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378.

For the purpose of institution and based on the record before us, we determine there is a reasonable likelihood Petitioner would prevail in showing that claims 9 and 10 are unpatentable under 35 U.S.C. § 103 over Khandekar, Le, and Sidebottom.

4. Challenge to Claim 17 over Khandekar, Le, and Stone
a. Overview of Stone (Ex. 1007)

Stone is directed to performance monitoring and control of server computers and applications in a distributed computer system. *See* Ex. 1007 ¶ 1. Stone discloses a multi-tiered service model of an e-business web service that provides tiers of service including web servers, application servers, and databases. *See id.* ¶¶ 28, 40, Figs. 1, 2. The tiered model organizes servers, applications, databases, security devices, and other components into tiers that correspond to the data flow for each service. *See id.* ¶ 37. Statistics collected and events monitored can also be organized by service and tier, allowing an automated management server to diagnose problems and relate them to service-level objectives. *See id.* Using the tiered model, availability problems or performance bottlenecks can easily be identified and corrected by automated management processes. *See id.* ¶ 38. A service-level-objective (SLO) monitoring and control system is provided that includes a SLO agent, service agents, node monitors, and local service agents. *See id.* ¶¶ 40–46, Fig. 2. The SLO agent has information about all

SLOs for all services and can best decide how to readjust priorities or allocate services among the available machines to optimize the SLOs. *See id.* ¶ 55. When SLO violations occur, the SLO agent can automatically adjust service priorities according to pre-configured policies without intervention by an administrator. *See id.* When the SLO agent needs to increase performance of a service, the SLO agent can search for less-utilized nodes. *See id.* ¶ 57. The nodes identified can be target nodes to which the applications are replicated when attempting to increase availability and performance of the service. *See id.* at code (57), ¶ 57.

b. Analysis

For the reasons that follow, we are persuaded that Petitioner establishes sufficiently, for the purpose of institution, that the combination of Khandekar, Le, and Stone teaches, suggests, or renders obvious the limitations recited in dependent claim 17.

Claim 17 depends from claim 1 and further recites “the application nodes are organized into tiers, and wherein status data is collected based on the tiers.” Ex. 1001, 37:53–55. Petitioner asserts that Khandekar explains its teachings may be applied to a web application that includes a web server, an application server, and a database server on a network of host computers. *See* Pet. 80 (citing Ex. 1004, 23:1–21). Petitioner contends that this three-tiered architecture was common for web sites. *See id.* (citing Ex. 1003 ¶ 259). Petitioner asserts that Khandekar teaches collecting status data of the hosts. *See id.* at 83 (quoting Ex. 1004, 4:61–63; citing Ex. 1004, 4:58–65, 12:57–13:8, 13:44–56, 23:28–51; Ex. 1003 ¶ 269). Petitioner contends that Le also teaches that data is collected. *See id.* (quoting Ex. 1006, 68:21–24; citing Ex. 1003 ¶ 270).

Petitioner asserts that Stone teaches the recited distributed computing system with “application nodes organized into tiers,” for example, a multi-tiered service model of an e-business web service, with web server, application, and database tiers. *See* Pet. 80–81 (reproducing Ex. 1007, Fig. 2 (with emphasis added); citing Ex. 1003 ¶ 260), 84 (reproducing Ex. 1007, Fig. 1; citing Ex. 1007 ¶¶ 1, 28–31, 34; Ex. 1003 ¶ 272). As to claim 17’s requirement that “status data is collected based on the tiers,” Petitioner asserts that Stone teaches an SLO monitoring and control system that collects statistics and monitors events by tier, which allows an automated management process to diagnose and correct problems if a service level objective is not met, such as by replicating nodes or increasing resources. *See id.* at 82 (citing Ex. 1007, code (57), ¶¶ 37, 38, 40, 57). Petitioner contends that Stone’s tiered service model allows automated diagnosis based on service level objectives and automatic reallocation within a tier. *See id.* at 81–82 (citing Ex. 1007 ¶¶ 37, 38, 55).

For the purpose of institution and based on Petitioner’s citations to Khandekar, Le, Stone, and Dr. Shenoy’s testimony (Ex. 1003), we determine Petitioner shows sufficiently that the Khandekar-Le-Stone combination teaches the limitations of claim 17. We are further persuaded Petitioner sets forth sufficient articulated reasoning with rational underpinning to support the conclusion that it would have been obvious to one of ordinary skill in the art to modify the combined teachings of Khandekar and Le to include organizing the application nodes into tiers and collecting status data based on the tiers, as taught by Stone, in order to allow automated diagnosis based on service-level objectives and allow automatic reallocation within a tier. *See* Pet. 81–82; *KSR*, 550 U.S. at 418. Petitioner contends a person of

ordinary skill in the art would have been motivated to apply Stone's tier-based teachings to improve the distributed computing system taught by the combination of Khandekar and Le. *See* Pet. 82 (citing Ex. 1003 ¶ 261). Petitioner contends that the combination is the application of well-known prior art elements according to known methods that yield predictable results. *See id.* at 83 (citing Ex. 1003 ¶ 266).

Patent Owner does not address substantively Petitioner's contentions based on the combination of Khandekar, Le, and Stone addressing the additional limitation of claim 17. *See* Prelim. Resp. 31. Nevertheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378.

For the purpose of institution and based on the record before us, we determine there is a reasonable likelihood Petitioner would prevail in showing that claim 17 is unpatentable under 35 U.S.C. § 103 over Khandekar, Le, and Stone.

III. CONCLUSION

For the foregoing reasons, we determine there is a reasonable likelihood that Petitioner would prevail in showing at least one of the challenged claims of the '138 Patent is unpatentable.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that, pursuant to 35 U.S.C. § 314(a), an *inter partes* review of claims 1–5, 9, 10, 17, 19–21, and 26 of the '138 Patent is instituted with respect to all grounds set forth in the Petition; and

IPR2020-01582

Patent 8,572,138 B2

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4(b), *inter partes* review of the '138 Patent shall commence on the entry date of this Order, and notice is hereby given of the institution of a trial.

IPR2020-01582
Patent 8,572,138 B2

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