

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC.,
Petitioner,

v.

CA, INC.,
Patent Owner.

IPR2022-00322
Patent 8,656,419 B2

Before JEAN R. HOMERE, THOMAS L. GIANNETTI,
and NATHAN A. ENGELS, *Administrative Patent Judges*.

HOMERE, *Administrative Patent Judge*.

DECISION
Final Written Decision
Determining All Challenged Claims Unpatentable
35 U.S.C. § 318 (a)

I. INTRODUCTION

A. *Background and Summary*

Netflix, Inc. (“Petitioner”) filed a Petition requesting *inter partes* review of claims 1–20 (“the challenged claims”) of U.S. Patent No. 8,656,419 B2 (Ex. 1001, “the ’419 patent”). Paper 2 (“Pet.”). CA, Inc. (“Patent Owner”) filed a Preliminary Response. Paper 7 (“Prelim. Resp.”). Upon considering the parties’ contentions and supporting evidence, the Board instituted an *inter partes* review pursuant to 35 U.S.C. § 314, as to the challenged claims of the ’419 patent with respect to all grounds set forth in the Petition. Paper 8 (“Inst. Dec.”).

After institution, Patent Owner filed a Response (Paper 13, “PO Resp.”); Petitioner filed a Reply to Patent Owner’s Response (Paper 16, “Pet. Reply”); Patent Owner filed a Sur-Reply to Petitioner’s Reply (Paper 18, “PO Sur-Reply”). At the request of both parties, we held an oral hearing on June 2, 2023. A transcript of the hearing has been entered into the record. Paper 25 (“Tr.”).

We have jurisdiction under 35 U.S.C. § 6. This Final Written Decision is entered pursuant to 35 U.S.C. § 318(a). For the reasons that follow, we determine that Petitioner has demonstrated by a preponderance of the evidence that challenged claims 1–20 of the ’419 patent are unpatentable.

B. Real Parties-in-Interest

Petitioner identifies itself and Netflix Streaming Services, Inc. as the real parties-in-interest. Pet. 76. Patent Owner identifies itself as the real party-in-interest. Paper 4, 1.

C. Related Matters

The parties indicate that the '419 patent is involved in the following civil actions and administrative proceedings:

CA, Inc. v. Netflix, Inc., 2:21-cv-00080 (E.D. Tex. 2021);

Netflix, Inc. v. CA, Inc., 3:21-cv-03649 (N.D. Cal. 2021);

CA, Inc. v. Netflix, Inc., 3:22-cv-00373 (N.D. Cal. 2021);

In re Netflix, Inc., 2022-110 (Fed. Cir. 2021); and

In re Netflix, Inc., 2021-0190 (Fed. Cir. 2021).

Pet. 76; Paper 4, 1.

D. The '419 Patent

The '419 patent relates to a method and system for distributed computing (e.g., remote procedure call allowing a client to execute an operation on a server). Ex. 1001, 1:5–13. As shown in Figure 1 below, the '419 patent describes dynamic distributed evaluator system 10. *Id.* at 2:1–2.

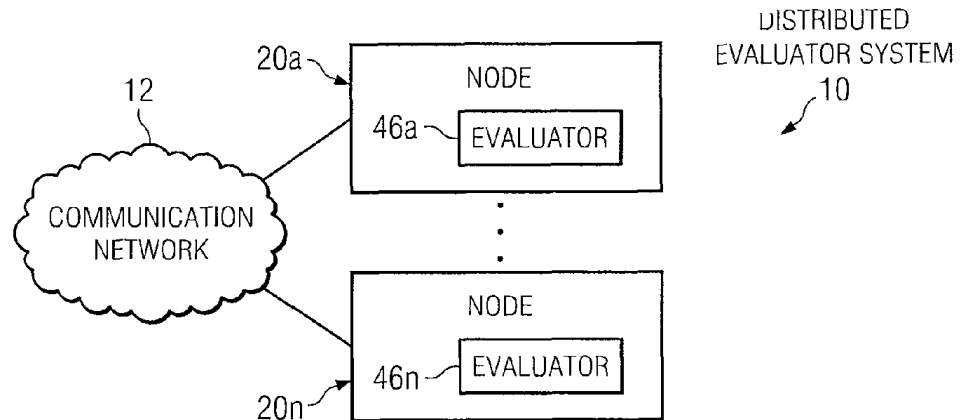


FIG. 1

Figure 1, reproduced above, depicts dynamic distributed evaluator system 10 including nodes 20a–n, each node including evaluator 46a–n for sending and receiving instructions, as well as for locating and transmitting a code to run an application containing various operations that may be performed by nodes 20a–n using the transmitted code. *Id.* at 2:2–12. In particular, first node 20a transmits via communication network 12 to second node 20n an operation including instructions along with a code instructing second node 20n how to perform the operation and what to do with the result of the operation. *Id.* at 2:26–33, 4:15–21, Fig. 3.

In another embodiment, first node 20a may transmit the instructions to second node 20n without knowing the code required to perform the underlying operation. *Id.* at 4:37–40. Alternatively, first node 20a may send an operation for distribution to the grid of nodes 20a–20n, without designating a particular node or code for performing the operation. *Id.* at 1:27–29.

E. Illustrative Claim

Of the challenged claims, claims 1, 12, and 18 are independent.

Claim 1 is illustrative and is reproduced below:

1. An apparatus, comprising a first node of a network, the first node comprising:
 - an interface operable to:
 - communicate with a second node of the network; and
 - one or more processors operable to:
 - tell a plurality of nodes to perform an operation comprising a procedure of an application, the plurality of nodes comprising a second node and one or more additional nodes; instruct the plurality of nodes how to perform the operation using computer code; and
 - tell the plurality of nodes what to do with a result of the operation, and
 - wherein the one or more processors does not know which one of the plurality of nodes will perform the operation.

Ex. 1001, 7:12–27.

F. Asserted Prior Art

Reference		Date	Exhibit
Jalan	US 2004/0172626 A1	Sept. 2, 2004	1005
Verbeke	US 2004/0261069 A1	Dec. 23, 2004	1004
Neiman	US 2006/0031842 A1	Feb. 9, 2006	1006

G. Asserted Grounds of Unpatentability

Petitioner asserts that the challenged claims are unpatentable on the following grounds:

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
1–4, 6–10, 12, 13, 15, 16, 18	103(a) ¹	Verbeke
5, 11, 14, 17, 19, 20	103(a)	Verbeke, Jalan
5, 11, 14, 17, 19, 20	103(a)	Verbeke, Jalan, Neiman

II. DISCUSSION

A. Claim Construction

We apply the same claim construction standard that would be used in a civil action under 35 U.S.C. § 282(b), following the standard articulated in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). 37 C.F.R. § 42.100(b). In applying such standard, claim terms are generally given their ordinary and customary meaning, as would be understood by a person of ordinary skill in the art, at the time of the invention and in the context of the entire patent disclosure. *Phillips*, 415 F.3d at 1312–13. “In determining the meaning of [a] disputed claim limitation, we look principally to the intrinsic evidence of record, examining the claim language itself, the written description, and the prosecution history, if in evidence.” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 469 F.3d 1005, 1014 (Fed. Cir. 2006) (citing *Phillips*, 415 F.3d at 1312–17).

¹ Because the claims at issue have an effective filing date prior to March 16, 2013, the effective date of the applicable provisions of the Leahy-Smith America Invents Act, Pub. L. No. 112-29, 125 Stat. 284 (2011) (“AIA”), we apply the pre-AIA version of 35 U.S.C. § 103 in this Decision.

Petitioner proposes to construe “a procedure of [an/the] application” as “a subpart of [an/the] application.” Pet. 20 (citing Ex. 1003 ¶¶ 62–67). Patent Owner characterizes the proposed claim construction as being “not material” to the arguments in its Response, and thereby declines to offer an alternative claim construction. PO Resp. 13. Based on the record before us, we do not discern a need to provide any express claim constructions because doing so would not change the outcome of the analysis below. *See 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.’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

B. Level of Ordinary Skill in the Art

In determining the level of ordinary skill in the art, various factors may be considered, including the “type of problems encountered in the art; prior art solutions to those problems; rapidity with which innovations are made; sophistication of the technology; and educational level of active workers in the field.” *In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995) (internal quotation and citation omitted). In that regard, Petitioner’s Declarant, Dr. Jagadish, testifies that a person with ordinary skill in the art (“POSITA”) for the ’419 patent

would have been a person with a bachelor’s degree in computer science, or a similar field, with at least two years of experience in the design of distributed computing systems or a person with a master’s degree in computer science, or a similar field, with a specialization in such systems.

Ex. 1003 ¶¶ 57–59.

Further, Petitioner contends that the level of ordinary skill could have varied somewhat to allow industry experience and education to compensate for each other. Pet. 19–20. Patent Owner contends to not “necessarily agree with Petitioner’s assertions regarding the level of ordinary skill”; however, Patent Owner declines to offer an alternative level of skill for the ordinary artisan as “any differences Patent Owner may have with Petitioner’s assertions are not material to the arguments in this Response.” PO Resp. 13.

On this record, we observe that Petitioner’s undisputed assessment of a person of ordinary skill in the art is consistent with the level of ordinary skill in the art at the time of the invention as reflected in the prior art in the instant proceeding. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001) (“[T]he level of skill in the art is a prism or lens through which a judge, jury, or the Board views the prior art and the claimed invention.”); *Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 718 (Fed. Cir. 1991) (“The importance of resolving the level of ordinary skill in the art lies in the necessity of maintaining objectivity in the obviousness inquiry.”). To that end, we note that the prior art itself often reflects an appropriate skill level. *See Okajima*, 261 F.3d at 1355.

Therefore, for purposes of this Decision, we adopt Petitioner’s assessment, except we decline to adopt “at least” as that language is vague and open-ended. Pet. 19. This assessment is consistent with the ’419 patent and the asserted prior art, and we apply it in our analysis below.

C. Obviousness over Verbeke

Petitioner asserts that claims 1–4, 6–10, 12, 13, 15, 16, 18 are unpatentable under 35 U.S.C. § 103(a) as obvious over Verbeke. Pet. 20–59. Petitioner also asserts that claims 5, 11, 14, 17, 19, and 20 are unpatentable under § 103(a) as obvious over the combination of Verbeke and Jalan, as well as the combination of Verbeke, Jalan, and Neiman. *Id.* at 59–75. Patent Owner opposes. PO Resp. 14–70. Upon considering Petitioner’s explanations and supporting evidence on the record before us, we are persuaded that Petitioner has demonstrated by a preponderance of the evidence that claims 1–20 are unpatentable as asserted above.

1. *Principles of Law*

“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) (citing 35 U.S.C. § 312(a)(3) (requiring *inter partes* review petitions to identify “with particularity . . . the evidence that supports the grounds for the challenge to each claim”)); *see also* 37 C.F.R. § 42.104(b) (requiring a petition for *inter partes* review to identify how the challenged claim is to be construed and where each element of the claim is found in the prior art patents or printed publications relied upon).

A patent 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 said

subject matter pertains. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved based on underlying factual determinations including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) when presented, objective evidence of nonobviousness, i.e., secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).²

Additionally, the obviousness inquiry typically requires an analysis of “whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.” *KSR*, 550 U.S. at 418 (*citing In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006) (requiring “articulated reasoning with some rational underpinning to support the legal conclusion of obviousness”)). Furthermore, Petitioner does not satisfy its burden of proving obviousness by employing “mere conclusory statements,” but “must instead articulate specific reasoning, based on evidence of record, to support the legal conclusion of obviousness.” *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016).

We analyze Petitioner’s obviousness grounds with the principles identified above in mind.

² Neither party presents evidence of secondary considerations of non-obviousness.

2. Overview of Verbeke

Verbeke describes a method and system for dynamically loading remote classes by a class loader without using separate custom class loaders. Ex. 1004, Abstract. In particular, Verbeke describes a remote class loader mechanism in a distributed computing system that locates remote system classes needed by an object-oriented code executing but not stored on a local system to remotely run the code on a Java Virtual Machine (JVM). *Id.*

¶¶ 13–14. Figure 2B of Verbeke is reproduced below.

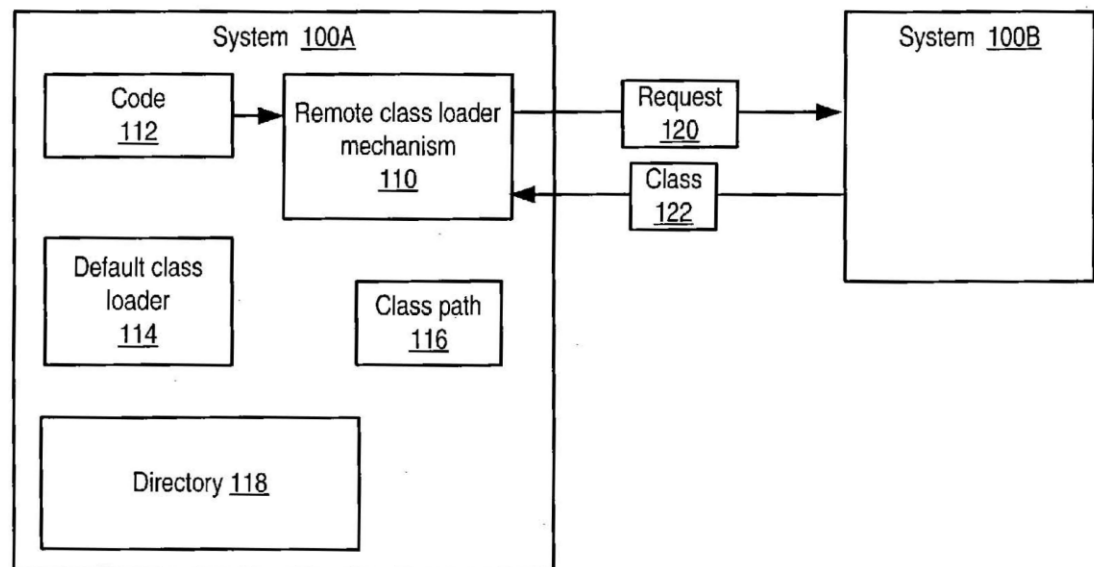


FIG. 2B

Figure 2B, reproduced above, depicts system 100A including remote class loader mechanism 110 for locating and downloading a class from a remote system 100B to execute code 112 of an application. *Id.* ¶ 56.

As shown in Figure 2B above, Verbeke discloses default class loader 114 in system 100A that looks in class path 116 to determine if class 122 is in local directory 118 to run code 112. *Id.* ¶¶ 55–56. Upon failing to find class 122 in directory 118, remote class loader mechanism 110 sends request 120 to remote system 100B (or other systems on a network) to locate desired class 122. *Id.* Verbeke further describes system 100B or another system in the network, upon identifying a local class file that satisfies the request, transmits class 122 to remote class loader mechanism 110. *Id.* ¶ 56.

Furthermore, Verbeke describes that downloaded code 112 may constitute an application fragmented at a framework (master node), which forwards application code fragments as tasks to different worker nodes to run the code fragments with associated classes stored locally or obtained from the master node. *Id.* ¶ 61. According to Verbeke, when the master node supplies a task (code fragment) for worker nodes in the distributed computing environment to run, the master node may not know all the classes the worker nodes need to run the application, and the worker nodes may not know anything about the application they are running. *Id.* ¶¶ 62, 63. Consequently, the worker nodes may need to use remote class loader mechanism 110 to download the necessary classes to run the application. *Id.* ¶ 62. Additionally, a top-level group may monitor and coordinate the overall activity of the framework including requests for peers to join the framework and their subsequent assignment node to peer groups. *Id.* ¶ 71.

Verbeke further discloses, upon receiving a job submission including numerous tasks, the task dispatcher may poll a repository peer group for the

status code within the code repository, and may distribute code and tasks to worker nodes to perform the assigned tasks and return the result to the task dispatcher, which in turn updates the repository with information about task completion status. *Id.* ¶¶ 74, 76, 77. Figure 6 of Verbeke is reproduced below.

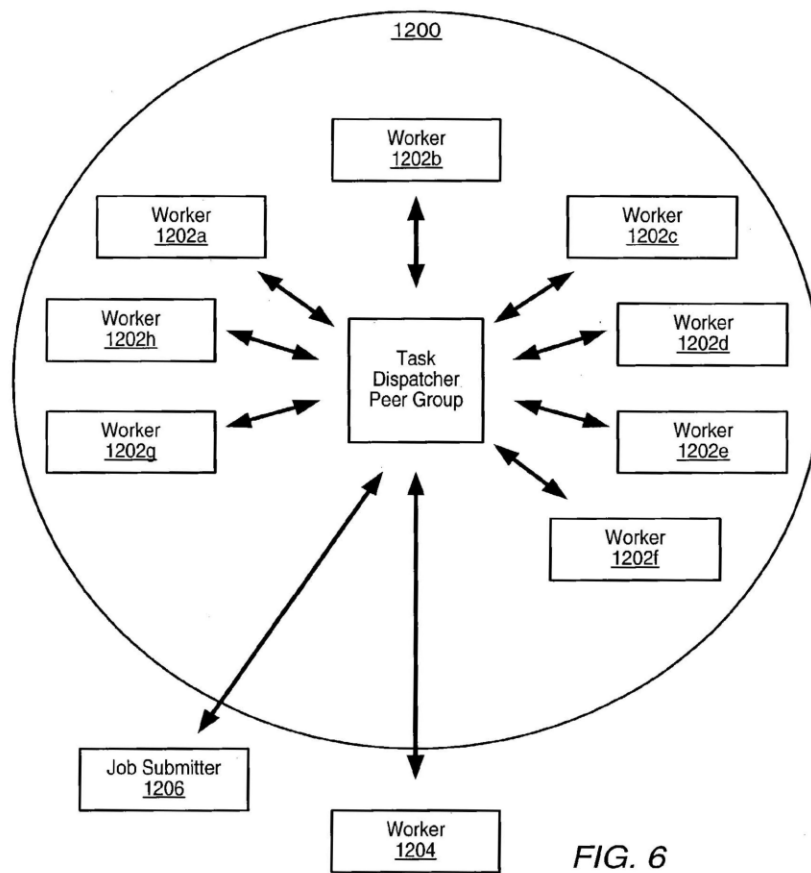


FIG. 6 depicts the task dispatcher, upon receiving a work assignment from job submitter 1206 (first node), dispatching received work assignment as code fragments/tasks to worker nodes 1202a–1202h of peer group 1200 to perform the assigned tasks and to return the results to the task dispatcher.

Id. ¶¶ 76, 81. Likewise, outside worker node 1204 may communicate to task dispatcher its desire to join the worker group. *Id.* ¶ 81.

3. *Obviousness Analysis*

Petitioner provides explanations to account for all of the claim limitations required by claims 1–4, 6–10, 12, 13, 15, 16, 18, citing Dr. Jagadish’s Declaration for support. Pet. 20–59; Ex. 1003. Petitioner contends that Verbeke teaches all limitations of the challenged claims, and that to the extent they are not explicitly disclosed, they would have been obvious under 35 U.S.C. § 103 to a person of ordinary skill in the art. Pet. 20.

Independent Claim 1

Petitioner’s Contentions

Independent claim 1 recites “[a]n apparatus, comprising.” Ex. 1001, 7:12. Petitioner contends that Verbeke discloses “a distributed computing system and method of utilization” throughout. Pet. 21 (citing Ex. 1004, *passim*). More particularly, Petitioner contends that Verbeke discloses “a system and method for submitting computational tasks in a distributed heterogeneous networked environment.” *Id.* (citing Ex. 1003 ¶¶ 72–74; Ex. 1004 ¶¶ 67, 69). According to Petitioner, to the extent the recited “apparatus” is considered a limitation, Verbeke’s distributed computing system teaches the apparatus. *Id.*

1[a]

Independent claim 1 further recites “a first node of a network, the first node comprising: an interface operable to: communicate with a second node of the network; and one or more processors operable to:”. Ex. 1001, 7:12–16. Petitioner contends that Verbeke discloses a job submitting node (e.g., job submitter 1206) communicating via wired/wireless network interfaces with one or more nodes in a framework of the network, wherein the nodes include processors operable to perform system functions. Pet. 21–23 (citing Ex. 1003 ¶¶ 76–84; Ex. 1004 ¶¶ 4, 16, 43, 44, 66–69, 71–76, 79, 81–84, 93, 108, 109, 112, 114). Petitioner contends that Verbeke’s job submitting node submitting tasks to the cluster or group of nodes for execution by any peer node teaches the first node. *Id.* at 23 (citing Ex. 1004 ¶¶ 69, 71, 83, 84, 108). According to Petitioner, “[a] POSITA would have understood Verbeke’s job-submitting node (job submitter 1206) is a ‘first node of a network’ because it is the node that submits a job to Verbeke’s distributed framework of peer nodes for execution (e.g., by other worker nodes 1202a–1202h).” *Id.* Further, Petitioner contends that Verbeke’s framework including worker nodes 1200a–1200h of task dispatcher peer group that utilize “wired or wireless network interfaces” to have a two-way communication with job submitting node 1206 teaches the first node comprising: an interface to communicate with a second node of a network and a third node of the network. *Id.* at 24–25 (citing

Ex. 1003 ¶¶ 85–87; Ex. 1004 ¶¶ 43, 44, 67, 69, 71, 74–76, 81, 82, 93).

Petitioner submits that although communications sent to a node framework are managed by a task dispatcher, a POSITA would have understood that such communications require an interface mechanism to establish two-way message exchange between any two elements in the network. *Id.* at 25–27 (citing Ex. 1003 ¶¶ 88–93; Ex. 1004 ¶¶ 44, 67, 72, 109). Additionally, Petitioner asserts that Verbeke’s disclosure of the peer nodes including the job-submitting node containing one or more processors teaches the recitation of “one or more processors operable to.” *Id.* at 27, 28 (citing Ex. 1003 ¶¶ 94–97; Ex. 1004 ¶¶ 67, 71, 109). Therefore, Petitioner submits that “[a] POSITA would have understood in 2008-2009 that distributing and running code requires the machine(s) have a processor of some type.” *Id.* at 27 (citing Ex. 1003 ¶ 95).

1[b]

Independent claim 1 further recites “tell a plurality of nodes to perform an operation comprising a procedure of an application, the plurality of nodes comprising a second node and one or more additional nodes.” Ex. 1001, 7:17–20. Petitioner contends that Verbeke’s framework including worker nodes 1202a–1202h for performing an operation containing tasks 1104a–1104c of a job assignment teaches a plurality of nodes comprising a second node and one or more additional nodes. Pet. 29–30 (citing Ex. 1003 ¶¶ 101–103). According to Petitioner, Verbeke’s disclosure of a processor at job submitter 1206 initiating an application containing tasks 1104a–1104c

to be performed by worker nodes 1202a–1202h teaches the processor at a first worker node telling a plurality of nodes of peer group 1200, including a second, third, and additional nodes to perform a plurality of task-making procedures of a job as instructed by the task dispatcher group. *Id.* at 30–31 (citing Ex. 1003 ¶¶ 103–105; Ex. 1004 ¶¶ 16, 60, 63, 64, 71–76, 81, 96, 101, 102). More particularly, Petitioner contends that because Verbeke’s task dispatcher distributes code fragments/tasks obtained from an application received from a submitting node for execution by worker nodes in a distributed computer framework, “a POSITA would have understood ‘a procedure of [an/the] application’ to be a subpart of an application.” *Id.* at 31–32 (citing Ex. 1003 ¶¶ 106, 107). Therefore, Petitioner contends that because “telling” and “sending instructions” are used interchangeably in the ’419 patent, a POSITA would have understood that Verbeke’s submitting tasks to a framework for distributed processing teaches instructing or telling a plurality of framework nodes to perform the identified operation or operations. *Id.* at 32–34 (citing Ex. 1003 ¶¶ 108–112). Further, Petitioner submits that although Verbeke’s submission of tasks is made to distributed framework (i.e., the tasks are sent to a node framework or pass through task dispatcher), a POSITA would have understood submitting tasks to a distributed framework is telling the plurality of nodes to perform operations because distributed processing is essential in a distributed system. *Id.* at 35 (citing Ex. 1003 ¶ 114).

1[c]

Independent claim 1 further recites “instruct the plurality of node how to perform the operation using computer code.” Ex. 1001, 7:21–22. Petitioner contends that Verbeke’s disclosure of a processor in work submitting node 1206 instructing worker nodes 1202a–1202h how to perform tasks 1104a–1104c in an operation using computer code teaches instructing the framework how to perform the submitted tasks using a particular code. Pet. 36–38 (citing Ex. 1003 ¶¶ 120–125; Ex. 1004 ¶¶ 60, 72, 74–76, 93, 96, 101, 102). According to Petitioner, a POSITA would have understood that Verbeke’s disclosure teaches instructing a plurality of nodes how to perform an operation because submitting identification code to a framework for distributed processing is tantamount to instructing or telling the nodes how to perform the tasks using the code. *Id.* at 40 (citing Ex. 1003 ¶¶ 112, 129–134).

1[d]

Furthermore, independent claim 1 recites “tell the plurality of nodes what to do with a result of the operation.” Ex. 1001, 7:23–24. Petitioner contends that Verbeke’s disclosure of job submitting node 1206 instructing worker nodes 1202a–1202h in the framework to perform tasks 1104a–1104c within a job request as well as to forward the task results back to a repository node managed by path dispatcher teaches a processor telling the framework what to do with the result of the operation. Pet. 41–43 (citing Ex. 1003 ¶¶ 138–140; Ex. 1004 ¶¶ 76, 79, 82, 96, 102). According to Petitioner, a POSITA would have recognized “by virtue of these instructions and the

creation of the job repository, the job submitter is also instructing the plurality of nodes what to do with the task results, at least because they are returned to the job repository defined by the job submitter's instructions.” *Id.* at 44 (citing Ex. 1003 ¶ 143). Therefore, Petitioner submits that a POSITA would have understood by 2008–2009 that a main reason to utilize a distributed computing system as in Verbeke's system was to perform distributed processing of an operation across worker nodes as well as to obtain a result of the distributed operation directly or indirectly. *Id.* at 44–45 (citing Ex. 1003 ¶¶ 144–152).

1[e]

Additionally, independent claim 1 recites “wherein the one or more processors does not know which one of the plurality of nodes will perform the operation.” Ex. 1001, 7:25–27. Petitioner contends that Verbeke's disclosure of job submitting node 1206 that submits a job request to task dispatcher node, which distributes subtasks 1104a–1104c to worker nodes 1202a–1202h in the framework teaches that job submitting node 1206 does not know which one of the nodes will perform the tasks. Pet. 46–48 (citing Ex. 1003 ¶¶ 155–156; Ex. 1004 ¶¶ 74, 76). According to Petitioner, because Verbeke's disclosure of using a unique ID is tied to the job as opposed to the node performing the operation, and tasks are tracked independently without identifying worker nodes to a job submitter, a POSITA would have expected that Verbeke's worker node anonymity to the job submitter would lead to well-known transparency in distributed systems. *Id.* at 46–47 (citing Ex. 1003 ¶¶ 157–160; Ex. 1004 ¶¶ 82, 144).

We have reviewed Petitioner's contentions and the underlying evidence and find those contentions to be persuasive. In particular, Petitioner's element-by-element (including the preamble) analysis for claim 1, as well as the specific references to Verbeke's teachings provide explanation and evidence that sufficiently support Petitioner's contentions. For the reasons discussed below, we do not agree with Patent Owner that the analysis set forth in the Petition fails to show that Verbeke teaches a first node telling a plurality of nodes to perform a same operation.

Patent Owner's Contentions that Limitations 1[b]–1[d] Allow No Indirect Communications Between the First Node and the Plurality of Nodes

With respect to the limitations 1[b]–1[d] set forth above, Patent Owner asserts that the record does not establish that these limitations allow for indirect communication, hierarchy, or “intermediation” between the first node and the plurality of nodes. PO Resp. 35. According to Patent Owner, the '419 patent instead discloses that separate communications including instructions may be sent substantially at the same time from the first node to the plurality of nodes, which may perform the instructions in parallel. *Id.* at 35–36 (citing Ex. 1001, 4:55–58, Ex. 2004 ¶¶ 64, 65). Therefore, Patent Owner asserts that Petitioner's reliance on limitation 1[e] to infer that limitations 1[b]–1[d] inherently include an intermediary is flawed because limitation 1[e] only requires that a processor is unaware of which node will perform the operation, as opposed to requiring anonymity of the nodes. *Id.*

at 36–37 (“Under the actual language of limitation [e], the first node can tell a plurality of nodes to perform an operation (and know exactly which nodes it told) but still not know which one will respond to the ‘telling.’”).

According to Patent Owner, the ’419 patent’s disclosure of “a technique by which the first node can communicate directly with the plurality of nodes without knowing which node will perform the operation,” requires that the first node communicates directly with the plurality of nodes, and needs not to be aware of which node will perform the operation. *Id.* at 37; PO Sur Reply 5–9.

Our Review

Based on the record before us, we disagree with Patent Owner’s contentions. We do not read the claim to require establishing a direct communication between the first node and the plurality of nodes, nor does the Specification expressly disclose such a direct communication. Because the cited claim language merely recites that the first node communicates with the plurality of nodes, the plain language of the claim encompasses both direct and indirect communications. As correctly argued by Petitioner, because Figure 1 of the ’419 patent depicts the first node communicating with the plurality of nodes via a communications network, which may include various devices (e.g., router, gateway), it encompasses indirect communication between the nodes. Pet. Reply 5–7 (citing Ex. 1031, 88:20–

89, 90:7–91: 9, 193:8–194:2; Ex. 1032 ¶ 3; Ex. 1001, Fig. 1)).³ Based on the record before us, we are persuaded by Petitioner’s conclusion that:

Verbeke further renders these obvious because it teaches its first node (e.g., job submitting node 1206) communicates over a network with a plurality of nodes of Verbeke’s framework including a node(s) of task dispatcher peer group, a node(s) of peer group 1200 including worker nodes 1202a-1202h, and a repository node(s), and that its nodes utilize “wired or wireless network interfaces.” *See, e.g.*, Ex-1004, ¶¶[0044], [0067], [0069], FIGS. 6, 8; Ex-1003, ¶85..

Pet. 24.

Patent Owner’s Contention that a First Node “tells” a Plurality of Nodes to Perform the Same, Single Operation

Patent Owner contends that Petitioner does not show that Verbeke’s disclosure teaches a first node to “tell” a plurality of nodes to perform an operation, as required by claim 1. PO Resp. 39. According to Patent Owner, a POSITA would understand that this plain claim language requires, at the very least, that “a plurality of nodes” *must be told to perform the operation*, irrespective of whether that is accomplished through direct or indirect communication. *Id.* at 39–40. Therefore, Patent Owner asserts that the language of the claim consistently supports a first node telling and

³ As noted by Petitioner, “CA’s direct communication argument is inconsistent with its district court infringement contentions, which rely upon indirect communication between the first node (highlighted in blue) and the second node (Titus Agents, highlighted in red).” *Id.* at 8–9 (citing Ex-1019, 000008, 000011).

instructing *the plurality of nodes* to perform *the same, single operation* (e.g., adding a new user to a computer) without knowing which of the nodes will perform the operation, whereas Verbeke teaches a job submitter that submits a job to a task dispatcher via a single communication, and the task dispatcher communicates each task included in the job to one of the worker nodes such that each worker node is assigned a different one of the tasks. *Id.* at 41–43 (citing Ex. 1001, 5:65–6:11; Ex. 2004 ¶¶ 73–75; Ex. 2005, 76:22–77:6, 82:16–21); PO Sur Reply 10–16.

Our Review

Based on the record before us, we disagree with Patent Owner’s contentions. As an initial matter, we note that although the disputed claim limitation requires that the first node tells the plurality of nodes to perform an operation, it does not require that the plurality of nodes perform the same operation on the same data subset. It suffices that the (plurality of) nodes are told by the first node to perform an operation (e.g. adding a user). Further, it is undisputed that Figure 5 Verbeke depicts a task dispatcher, upon receiving a job from a job submitter, dispatches partitions of the job to a plurality of worker nodes as a plurality of tasks such that each worker node is assigned a same operation (e.g., X-ray generation in aluminum) for a different job partition/task/data set (e.g., electrons 1 to 1200, electrons 201 to 1400, electrons 401 to 600). Pet. Reply 9–15 (citing Ex. 2004 ¶ 76; Ex.1031, 76:17–77:6, 81:22–82:7, 83:4–14, 84:17–85:5, 86:5–15, 184:8–185:11, 186:11–13, 19–187:19, 193:8–194:2; Ex.1019, 000008, 000011; Ex. 1032 ¶¶ 7, 8). In particular, as correctly noted by Petitioner, Patent Owner’s

expert acknowledges that “Figure 5 is an example in Verbeke where the task dispatcher tells or instructs three separate workers to perform the same task” *Id.* at 13.

We therefore agree with Petitioner that Verbeke’s disclosure of the task dispatcher assigning a job in different partitions/tasks to different worker nodes such that each worker node performs the same and single operation of Monte-Carlo X-ray generation in aluminum, albeit on partitioned sets or subsets of data, teaches or suggests a first node telling a plurality of nodes to perform a same, single operation. *Id.* After considering the parties’ evidence and arguments, we find Petitioner has shown by a preponderance of the evidence that Verbeke discloses the claimed subject matter in Element 1[b].

Patent Owner’s Contentions Regarding Telling a Plurality of Nodes to Perform an Operation

First, Patent Owner contends that Petitioner fails to establish that Verbeke teaches a first node telling a plurality of nodes to perform an operation, as required by claim 1. PO Resp. 44. In particular, Patent Owner asserts that because Verbeke’s job submitter communicates with the worker nodes via an intermediary task dispatcher, the job submitter does not tell the plurality of nodes to do anything. *Id.* (citing Ex. 2004 ¶¶ 78, 79). That is, because the job submitter submits a job for execution to the task dispatcher, which partitions the job into different tasks or code fragments before distributing it to the worker nodes, the latter nodes do not receive any

communication or instructions from the job submitter. *Id.* at 45–46 (citing Ex. 2004 ¶¶ 80–83).

According to Patent Owner, the communication between the job submitter and the task dispatcher is separate and different from the communications between the task dispatcher and the worker nodes. *Id.* at 46–49 (citing Ex. 2004 ¶¶ 83–86). More particularly, Patent Owner contends that because the job dispatcher, upon receiving the job from the job submitter, generates different messages to each of the workers, Verbeke does not suggest that the job submission sent by the job submitter is ever forwarded to any of the worker nodes. *Id.* at 49; PO Sur Reply 17–18.

Second, Patent Owner argues that Verbeke does not teach the task dispatcher telling multiple worker nodes to perform the same task; instead the task dispatcher communicates a different task for each worker to perform. PO Resp. 49–50 (citing Ex. 1004 ¶ 69; Ex. 2004 ¶¶ 87). According to Patent Owner, “the entire point” of the Verbeke’s distributed system is that while the task dispatcher may receive multiple tasks from the job submitter, the task dispatcher determines the most appropriate single worker for each task based on workload and code availability. *Id.* at 51–54 (citing Ex. 1004 ¶¶ 76, 96, 102; Ex. 2004 ¶ 88–94); PO Sur Reply 18–19.

Third, Patent Owner contends that Petitioner’s characterization of Verbeke’s task dispatcher as a multipurpose node, serving simultaneously as both a task dispatcher and as one of worker nodes is not supported by any evidence. PO Resp. 54–55. Patent Owner asserts that because neither the job submitter nor the task dispatcher communicates with the plurality of

nodes about an operation, Petitioner's multi-purpose node theory does not solve this issue. *Id.* at 56–57 (citing Ex. 2004 ¶¶ 97–100). According to Patent Owner, because the task dispatcher plays the role of determining which of the worker nodes a task should be assigned to, it cannot also play the role of a worker node. *Id.* at 57–58; PO Sur Reply 19–20.

Our Review

Based on the record before us, we disagree with Patent Owner's contentions. First, we do not agree with Patent Owner's contention that the job submitted to the task dispatcher by the job submitter is not communicated to the plurality of nodes. As correctly noted by Petitioner, the disputed claim limitations do not recite a "communication" or require a "single communication sent to the plurality of nodes." Pet. Reply 9. Instead, they merely require that the first node tell or instruct the plurality of nodes to perform an operation. *Id.* Although Verbeke's task dispatcher does not assign an entire job to one of the worker nodes, it assigns the aggregated tasks included in the job to the plurality of nodes. We therefore agree with Petitioner that, as admitted by Dr. Rosenblum, the claim requirement is met by Verbeke's sending different messages and associated tasks included in the submitted job to each of the nodes. *Id.* at 10 (citing Ex. 2004 80, Ex. 1031, 84:17–85:5; 86:5–15). More particularly, we agree with Petitioner that even though each node in the plurality of nodes does not receive the submitted job in its entirety, it suffices that the nodes as a whole receive the aggregated job to meet the disputed limitations. *Id.* at 10–11.

Second, we do not agree with Patent Owner that Verbeke's task dispatcher only tells a single worker node as opposed to a plurality of nodes to perform an operation. As noted in our discussion above, although the claim requires that the nodes perform the same operation (e.g., adding a user), it does not require that they perform the same operation on the same dataset. We therefore agree with Petitioner that Verbeke's task dispatcher telling each of the plurality of nodes to perform the same operation of generating aluminum, albeit on different tasks/dataset, teaches or suggests a first node telling a plurality of nodes to perform the same single operation. *Id.* at 11–15.

Third, we do not agree with Patent Owner's argument that neither Verbeke's job submitter nor the task dispatcher tells or instructs a plurality of nodes to perform an operation. Notwithstanding Petitioner's alleged multipurpose node theory referenced above, we reiterate our position that Verbeke's job submitter does tell or instruct the plurality of nodes (albeit indirectly via the task dispatcher) to perform a same single operation. In other words, because the task dispatcher relays to the plurality of worker nodes the tasks included in a job received from the job submitter such that each of the worker nodes is assigned to process one of the tasks, we agree with Petitioner that Verbeke teaches or suggests a first node telling a plurality of nodes to perform an operation as required by the disputed claim limitations. *Id.* at 15.

Patent Owner's Contentions Regarding "Instructing" a Plurality of Nodes How to Perform the Operation Using Computer Code

Patent Owner contends that Petitioner has not demonstrated that Verbeke teaches or suggests the job submitter instructing a node how to perform the operation using computer code. PO Resp. 58. In particular, Patent Owner asserts that Petitioner mischaracterizes Verbeke's disclosure of the job submitter providing an identified code to the task dispatcher, upon request, to instruct worker nodes to perform the tasks for the submitted job. *Id.* at 59 (citing Ex. 1003 ¶ 125; Ex. 1004 ¶ 60; Ex. 2004 ¶ 102). According to Patent Owner, although Verbeke's job submitter identifies and provides the code upon receiving a request from the task dispatcher, which subsequently forwards the code to the worker nodes to perform the job, the job submitter does not instruct the plurality of nodes how to perform the operation using the code; rather, the task dispatcher does. *Id.* at 59–61 (citing Ex. 1004 ¶¶ 101, 102, Ex. 2004 ¶ 103); PO Sur Reply 19–20.

Our Review

Based on the record before us, we disagree with Patent Owner's contentions. As correctly noted by Petitioner, Verbeke discloses a job submitting node submitting a job request along with an identified code to a task dispatcher, which assigns different portions of the job along with associated computer code to worker nodes for distributed processing. Pet. Reply 16–17 (citing Ex. 1004 ¶¶ 75, 96). Because the submitted code contains information as to how to perform the function and what to do with

the results, such a submission from the submitting node to the task dispatcher, which in turn forwards the code to the worker nodes, teaches or suggests the submitting node instructing the worker nodes how to perform the tasks using the code. As noted above, because the plain language of the claim does not require the first node to have direct communication with the plurality of nodes, we agree with Petitioner that Verbeke's disclosure of a communication (albeit indirect) between the submitter node and the worker nodes via the task dispatcher teaches the cited claim limitation. *Id.* at 18. This conclusion is uncontroverted by Dr. Rosenblum's testimony, which admits that Verbeke's Figure 5 describes a plurality of nodes that, upon receiving instructions from another node, execute a single task using a same code. Ex, 1031, 186:19–188:4. Accordingly, we are persuaded by Petitioner's conclusion that:

Verbeke repeatedly teaches its job submitter instructs the framework how to perform the tasks it submits using particular code, as-identified and/or as submitted. A POSITA would have understood this to be instructing a plurality of nodes how to perform an operation, as recited in 1[c], 12[c], and 18[c], because submitting identification of code or code to a framework for distributed processing is instructing or telling a plurality of nodes in the framework how to perform tasks using the code—that is the purpose of submitting tasks/code to a distributed system. Ex-1003, ¶128.

Pet. 40.

Patent Owner's Contentions Regarding Telling the Nodes What to do with the Operation Result

With respect to the limitation of claim 1 of “tell the plurality of nodes what to do with a result of the operation,” Patent Owner contends that Petitioner mischaracterizes Verbeke’s disclosure in asserting that the job submitter instructs the framework to return the task results back to repository nodes managed by the task dispatcher after the tasks are run by the worker nodes. PO Resp. 61–62 (citing Ex. 1004 ¶¶ 76, 79, 82, 96, 102; Ex. 2004 ¶ 105). According to Patent Owner, “the worker node never communicates with the repository—rather, the worker simply returns the result to the task dispatcher.” *Id.* at 62. Further, according to Patent Owner, although Verbeke’s task dispatcher instructs the worker nodes to store operation results in the repository from which they are retrieved by the task dispatcher upon receiving a request to do so by the job submitter, Verbeke’s job submitter does not tell the worker nodes what to do with the result; rather the task dispatcher does. *Id.* at 63–65; PO Sur Reply 20–23.

Our Review

Based on the record before us, we disagree with Patent Owner’s contentions. Paragraph 82 of Verbeke indicates that, upon receiving an indication from the worker nodes that all the assigned tasks in its repository are completed, the task dispatcher returns the results back to the job submitter, which in certain instances may *initiate the result retrieval process*. Ex. 1004 ¶ 82 (emphasis added).

As correctly noted by Petitioner, Verbeke discloses a job submitting node submitting a job request along with an identified code to a task

dispatcher, which assigns different portions of the job along with the computer code to worker nodes for distributed processing thereby instructing the worker nodes how to perform the tasks using the code. Pet. Reply 19 (citing Ex. 1004 ¶ 96). As further correctly noted by Petitioner, Verbeke also discloses the task dispatcher instructing the worker nodes to store the task results in a repository from which the task dispatcher retrieves and returns the results to the job submitter, which initiated the initial result retrieval process. *Id.* (citing Ex. 1004 ¶ 82).

As previously noted in our Institution Decision, and reiterated above, because the plain language of the claim does not require the first node to have direct communication with the plurality of nodes, we agree with Petitioner that Verbeke's disclosure of a communication (albeit indirect) between the submitter node and the worker nodes via the task dispatcher teaches the cited claim limitation. *Id.* at 20–21 (citing Inst. Dec. 25–26). Accordingly, we are persuaded, based on the record before us, by Petitioner's conclusion that:

[A] POSITA would have recognized that, by virtue of these instructions and the creation of the job repository, the job submitter is also instructing the plurality of nodes what to do with the task results, at least because they are returned to the job repository defined by the job submitter's instructions. Ex-1003 ¶143.

Pet. 44.

[A] POSITA would have understood by 2008-2009 a main reason to utilize a distributed computing system like Verbeke's was to obtain a result of a distributed operation—either directly or through an intermediate storage scheme, such as a log file or Verbeke's

repositories. *See* §I. In other words, it would have been obvious to instruct a distributed system made up of a plurality of nodes to return a result of a job submitted to the system back to the job submitter in view of a POSITA's knowledge, as that is the purpose of submitting the job—obtaining a result.

Id. at 45.

Furthermore, as correctly noted by Petitioner, Dr. Rosenblum's testimony acknowledges that it would have been obvious to tell the nodes what to do with the results of the operation. Pet. Reply 21–22 (citing Ex. 1031, 30:9–16, 34:15–35:7, 132:13–19, 32:12–33:21, 35:8–36:1; Ex. 1032 ¶ 10). Such an admission from Patent Owner's expert supports Petitioner's rationale to modify Verbeke in the suggested manner, and thereby weighs against Patent Owner's arguments to the contrary. Sur Reply 23.

Based on the evidence on the record before us, we find that Petitioner has demonstrated persuasively that Verbeke discloses the subject matter of the preamble and claim limitations 1[a]–1[e]. In light of the foregoing, we determine that Petitioner has demonstrated by a preponderance of the evidence that claim 1 is unpatentable under § 103(a) as obvious over Verbeke.

Independent Claims 12 and 18

Petitioner contends that claim 12 recites a method and claim 18 recites an apparatus substantially similar to that of claim 1, and relies upon a substantially similar analysis to show that claims 12 and 18 are unpatentable. Pet. 21–48. Petitioner relies on the same arguments and evidence presented

for claim 1 and provides additional, persuasive arguments and evidence in support of its position for slight differences in the claims. *Id.* Patent Owner does not present arguments in addition to those discussed with respect to claim 1 separately contesting Petitioner's assertions for claims 12 and 18. *See generally* PO Resp.

For the reasons given with respect to claim 1 above, based on the complete record now before us, we conclude that Petitioner has similarly established by a preponderance of the evidence that independent claims 12 and 18 are unpatentable under 35 U.S.C. § 103(a) as obvious over Verbeke.

Dependent Claims 2–4, 6, 10, and 13,

Petitioner's Contentions

Claims 2–4, 6, and 10 depend on independent claim 1. Ex. 1001, 7:27–44, 49–52, *id.* at 7:64–8:5. Claim 13 depends on independent claim 12. *Id.* at 8:28–38. Dependent claims 2–4, 6, 10, and 13 further expand upon the recitation of independent claims 1 and 12 such that the first node provides a code including the code identification, location, and instructions to tell the plurality of nodes when and how to perform an operation, as well as what to do with the result of the operation. *Id.*

Petitioner builds upon its analysis for independent claims 1 and 12 to particularly explain how Verbeke teaches the additional limitations of claims 2–4, 6, 10, and 13. Pet. 48–56, 58, 59 (citing Ex. 1003 ¶¶ 127, 163–229; Ex. 1004 ¶¶ 16, 46, 49, 52–65, 69, 74, 75, 82, 96, 101, 108, 145, 147).

Patent Owner's Response

Patent Owner relies on similar arguments as those discussed above for independent claims 1 and 12 1, and does not separately argue dependent claims 2–4, 6, 10, and 13. PO Resp. 67–68, and 70. Patent Owner does not present specific arguments regarding these claim elements.

Our Review

Based on the record before us, we agree with Petitioner that Verbeke teaches the limitations of dependent claims 2–4, 6, 10, and 13. In particular, we agree with Petitioner that Verbeke's disclosure of a job submitter submitting a code to its framework for distribution to worker nodes teaches the first node sending the code including its identification and location to the plurality of nodes, albeit via its framework, to instruct the plurality of nodes how to perform an operation, as well as what to do the result of the operation. Pet. 48–56, 58, 59. More particularly, we agree with Petitioner that a POSITA would have understood that Verbeke's submission of a code to its framework for distribution to the plurality of nodes to perform a task teaches or suggests the additional limitations of dependent claims 2–6, 10, and 13. *Id.* We have reviewed Petitioner's arguments and the underlying evidence cited in support thereof, and we determine that Petitioner has demonstrated by a preponderance of the evidence that claims 2–4, 6, 10, and 13 are unpatentable under § 103(a) as obvious over Verbeke.

Dependent Claims 7, 8, and 15

Petitioner's Contentions

Claims 7 and 8 depend on independent claim 1. Ex. 1001, 7:53–55. Claim 15 depends on independent claim 12. *Id.* at 8:43–45. Dependent claims 7, 8, and 15 further expand upon the recitation of independent claims 1 and 12 such that the code is programmed to remove itself upon completion of a process or the expiration of a timer. *Id.*

Petitioner builds upon its analysis for independent claims 1 and 12 to particularly explain how Verbeke teaches the additional limitations of claims 7, 8, and 15. Pet. 55–56 (citing Ex. 1003 ¶¶ 206–211; Ex. 1004 ¶¶ 145, 147).

Patent Owner's Response

Patent Owner contends that Petitioner has not demonstrated that Verbeke teaches “code programmed to remove itself upon the completion of a process or upon the expiration of a timer (e.g., when a job or tasks are completed) and instructing the framework to do the same,” as recited in claims 7 and 15. PO Resp. 68; PO Sur Reply 24–25. In particular, Patent Owner contends the following:

Petitioner misreads Verbeke's `remove()` and `quit()` methods as disclosing the removal of code from a worker. *See* Pet., 55; EX-2004, ¶¶ 117–118. Instead, Verbeke specifically teaches that “[t]he `quit()` method [is] called before exiting the application to quit cleanly” while “`remove()` [simply] removes all tasks belonging to this job from the task repository” to prevent “memory requirements of the repository manager” from increasing over time...

No POSITA would understand either of these functions as even suggesting that code might remove itself from a worker. EX-2004, ¶118. To the contrary, Verbeke implicitly teaches that code remains on a worker node after completion of the task, and Verbeke indicates that this is beneficial, because the worker is able to perform a future task with the same code more efficiently...

PO Resp. 68–69 (citing Ex. 1004 ¶¶ 145, 147; Ex. 2004 ¶¶ 117–119).

Our Review

Based on the record before us, we disagree with Patent Owner’s contentions. As correctly noted by Petitioner, the claim limitation merely requires the code programmed to remove itself upon completion of a process without specifying that such removal is necessarily from a worker node.

Pet. Reply 23. We agree with Petitioner that a POSITA would readily recognize that Verbeke’s disclosure of the remove() method to remove all tasks belonging to a job from a tasks repository upon completion of the tasks teaches the disputed claim limitation. *Id.* (citing Ex. 1004 ¶ 147).

Furthermore, we agree with Petitioner that

a POSITA would have found it obvious to similarly remove code from a worker node to prevent the memory requirements of the worker node from increasing over time... Ex-1032.

a POSITA would have understood “cached” to mean storing in a cache memory, which is a small temporary memory reserved for the most recently or frequently used code, and is frequently cleared to make space for the next code to be run.

Id. at 24 (citing Ex.1032 ¶¶ 12, 13).

We have reviewed Petitioner’s arguments and the underlying evidence cited in support thereof , and we determine that Petitioner has demonstrated by a preponderance of the evidence that claims 7, 8, and 15 are unpatentable under § 103(a) as obvious over Verbeke.

Dependent Claims 9 and 16

Petitioner’s Contentions

Claim 9 depends on independent claim 1. Ex. 1001, 7:58–63. Claim 16 depends on independent claim 12. *Id.* at 8: 46–51. Dependent claims 9 and 16 further expand upon the recitation of independent claims 1 and 12 such that the first node instruct a plurality of nodes to use a first code and a second code comprising a first programming language and a second programming language, respectively. *Id.*

Petitioner builds upon its analysis for independent claims 1 and 12 to particularly explain how Verbeke teaches the additional limitations of claims 9 and 16. Pet. 56–57 (citing Ex. 1003 ¶¶ 212–220; Ex. 1004 ¶¶ 46, 60, 74, 96).

Patent Owner’s Response

Patent Owner contends that Petitioner has not demonstrated that Verbeke teaches “instructing the second node to use a first code comprising a first programming language and second code comprising a second programming language,” as recited in dependent claims 9 and 16. PO Resp. 69 (citing Ex. 1001, 7:61–63, 8:49–51). In particular, Patent Owner

contends that Verbeke's disclosure of code that "may be an application, applet, servlet, executable code fragment, or any other executable code" and "may be written in a bytecode-level language recognized by all or a subset of the systems in a network (e.g., Java programming language) does not teach or suggest the claim limitation. *Id.* (citing Ex. 1004 ¶ 46). According to Patent Owner, a POSITA would readily recognize that the types of code disclosed by Verbeke are directed to the Java language, and would not any functionality outside of the Java specific environment. *Id.* at 69–70 (citing Ex. 2004 ¶ 121); Sur Reply 25.

Our Review

Based on the record before us, we disagree with Patent Owner's contentions. As correctly noted by Petitioner, although Verbeke discloses an embodiment with sample code written in Java, it also indicates that alternative embodiments written in other programming languages are within the purview of the disclosed reference. Pet. Reply 25 (citing Ex. 1004 ¶ 140). Accordingly, we agree with Dr. Jagadish's testimony, unconverted by Dr. Rosenblum, that:

[A] POSITA would have understood by 2008-2009 that distributed systems may run tasks on individual nodes in different programming languages.

[A] POSITA in 2009 would have had an understanding of contemporary programming languages of the time, possibly C, C++, Java, and some of the concepts around deploying programs in distributed systems, programs that were written in those languages.

Id. at 25–26 (citing Ex. 1003 ¶¶ 215–217; Ex. 1014, 6:41–45, Ex. 1006 ¶ 49; Ex. 1031, 28:14–29:13).

We have reviewed Petitioner’s arguments and the underlying evidence cited in support thereof, and we determine that Petitioner has demonstrated by a preponderance of the evidence that claims 9 and 16 are unpatentable under § 103(a) as obvious over Verbeke.

Dependent Claims 5, 11, 14, 17, 19, and 20

Petitioner’s Contentions

Dependent claims 5 and 11 depend on independent claim 1. Ex. 1001, 7:45–48, 8:6–15. Claims 14 and 17 depend on independent claim 12. *Id.* at 8:3943, 52–58. Claim 19 and 20 depend on independent claim 18. *Id.* at 9:10–19. Dependent claims 5, 11, 14, 17, 19, and 20 further expand upon the recitation of independent claims 1, 12, and 18 such that the first node tell the plurality of nodes or the second node to send the result of the operation to a third node. *Id.*

Petitioner builds upon its analysis for independent claims 1, 12, and 18, to particularly explain how the combination of Verbeke with Jalan, as well as the combination of Verbeke with Jalan, and Neiman teaches the additional limitations of claims 5, 11, 14, 17, 19, and 20. Pet. 59–75 (citing Ex. 1003 ¶¶ 230–293; Ex. 1004 ¶¶ 2–12, 16, 66–82, 93, 96, 176; Ex. 1005 ¶¶ 4, 11, 12, 23–26, 98, 99, 188, 200–204; Ex. 1006 ¶¶ 15, 16, 37–41, 66, 67, 116–120).

Patent Owner's Response

Patent Owner relies on similar arguments as those discussed above for independent claims 1, 12 and 18, and does not separately argue dependent claims 5, 11, 14, 17, 19, and 20. PO Resp. 70. Patent Owner does not present specific arguments regarding these claim elements.

Our Review

Based on the record before us, and, we agree with Petitioner Verbeke teaches the limitations of dependent claims 5, 11, 14, 17, 19, and 20. In particular, we agree with Petitioner that Verbeke's disclosure of a job submitter submitting a code to its framework for distribution to worker nodes teaches the first node sending the code to the plurality of nodes, albeit via its framework, to instruct the plurality of nodes how to perform an operation, as well as what to do the result of the operation. Pet. 60. We further agree with Petitioner that Jalan's parallelization and Neiman's job division would complement Verbeke's teaching to predictably result in a distributed computing system, which processes a job more efficiently between a plurality of nodes. *Id.* at 60–61.

We have reviewed Petitioner's arguments and the underlying evidence cited in support thereof, and we determine that Petitioner has demonstrated by a preponderance of the evidence that claims 5, 11, 14, 17, 19, and 20 are unpatentable under § 103(a) as obvious over the combination of Verbeke with Jalan, as well as the combination of Verbeke with Jalan, and Neiman.

III. CONCLUSION

The outcome for the challenged claims is set forth below.⁴ In summary:

Claims Challenged	35 U.S.C. §	Reference(s)/ Basis	Claims Shown Unpatentable	Claims Not Shown Unpatentable
1–4, 6–10, 12, 13, 15, 16, 18	103(a)	Verbeke	1–4, 6–10, 12, 13, 15, 16, 18	
5, 11, 14, 17, 19, 20	103(a)	Verbeke, Jalan,	5, 11, 14, 17, 19, 20	
5, 11, 14, 17, 19, 20	103(a)	Verbeke, Jalan, Neiman	5, 11, 14, 17, 19, 20	
Overall Outcome			1–20	

Accordingly, based on the complete record now before us, we determine that Petitioner has demonstrated by a preponderance of the evidence that claims 1–20 are unpatentable as obvious under § 103(a) over

⁴ Should Patent Owner wish to pursue amendment of the challenged claims in a reissue or reexamination proceeding subsequent to the issuance of this decision, we draw Patent Owner’s attention to the April 2019 *Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding*. See 84 Fed. Reg. 16,654 (Apr. 22, 2019). If Patent Owner chooses to file a reissue application or a request for reexamination of the challenged patent, we remind Patent Owner of its continuing obligation to notify the Board of any such related matters in updated mandatory notices. See 37 C.F.R. §§ 42.8(a)(3), (b)(2).

Verbeke, Verbeke in combination with Jalan, and Verbeke in combination with Jalan and Neiman, as set forth above.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that Petitioner has shown by a preponderance of the evidence that claims 1–20 of the '419 patent are unpatentable.

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.

IPR2022-00322
Patent 8,656,419 B2

For PETITIONER:

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

For PATENT OWNER:

Daniel S. Young
ADSERO IP LLC D/B/A/
SWANSON & BRATSCHUN LLC
dyoung@adseroip.com
dyoung@sbiplaw.com

Chad E. King
KING IAM, LLC
chad@king-iam.com