

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

UNIFIED PATENTS, LLC,
Petitioner,

v.

VALTRUS INNOVATIONS LIMITED,
Patent Owner.

IPR2022-01399
Patent 7,748,005 B2

Before NORMAN H. BEAMER, SHARON FENICK,
and PAUL J. KORNICZKY, *Administrative Patent Judges*.

BEAMER, *Administrative Patent Judge*.

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

I. INTRODUCTION

On August 10, 2022, Unified Patents, LLC (“Petitioner”) filed a Petition (“Pet.”) pursuant to 35 U.S.C. §§ 311–319 to institute an *inter partes* review of claims 8–14 of U.S. Patent No. 7,748,005 B2 (Ex. 1001, “the ’005 patent”). Paper 2. On January 30, 2023, Valtrus Innovations Limited (“Patent Owner”) filed a Preliminary Response (“Prelim. Resp.”). Paper 6. As authorized by the Board, Petitioner filed a Reply and Patent Owner filed a Sur-Reply directed to discretionary denial issues. Ex. 1022; Papers 7, 8.

The standard for instituting an *inter partes* review is set forth in 35 U.S.C. § 314(a), which provides that an *inter partes* review may not be instituted unless the information presented in the Petition and any preliminary response 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.”

For the reasons explained below, we deny the Petition and do not institute an *inter partes* review.

II. BACKGROUND

A. The ’005 Patent

The ’005 patent, titled “System And Method For Allocating A Plurality Of Resources Between A Plurality Of Computing Domains,” was filed on September 10, 2004, issued on June 19, 2010, and is a continuation-in-part of application No. 10/206,594, filed on July 26, 2002, now U.S. Patent No. 7,140,020, which is a continuation-in-part of application No. 09/493,753, filed on January 28, 2000, now U.S. Patent No. 7,228,546. Ex. 1001, codes (54), (22), (45), (63).

The '005 patent is directed to allocating resources to computing domains. Ex. 1001, 1:19–21. The '005 patent specifically describes two types of resource allocations to computer domains: “allocating physical resources between partitions,” and allocating “virtual resources as software constructs” to “virtual machines.” *Id.* at 13:37–47. Resource allocation is performed dynamically by the system, in response to changes in workload characteristics therein, based on performance goals with different priorities defined by an administrator. *Id.* at 3:6–7, 3:14–20, 4:36–40.

For physical partitions, the '005 patent explains that computer systems, which can be “a single small computer . . . a single large computer . . . or a network of larger and/or small computers,” can be divided into “protection domains or partitions.” Ex. 1001, 3:7–14. “Each partition may have one or more processors and other systems resources, e.g. storage devices, I/O devices, etc.” *Id.* at 4:61–63. “Each partition is preferably running its own operating system . . . which provides segregation and survivability between the partitions.” *Id.* at 4:63–66.

For virtual machines, the '005 patent explains that:

[V]irtualization refers to the creation of virtual machines that coexist on one or several physical servers. Virtualization software typically executes in connection with a host operating system of the physical server. The virtualization software creates virtual resources as software constructs. The virtual resources are then assigned to virtual machines used for respective servers. Specifically, the virtual resources are used to execute “guest” operating systems that execute on top of the host operating system. The guest operating systems are then used to execute applications. Furthermore, each guest operating system operates independently. A software fault associated

with any particular guest operating system and its application(s) may be contained within a given virtual machine.

Ex. 1001, 13:42–55; *see also id.* at 1:37–48.

Figure 1, reproduced below, illustrates a physical partition embodiment of the '005 patent subject matter.

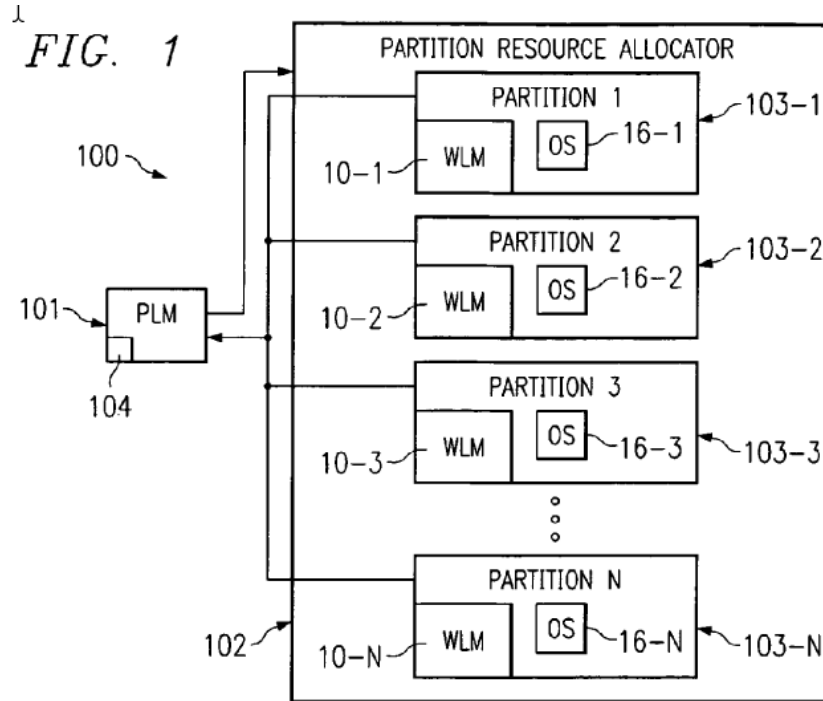


Figure 1 depicts partition load manager (PLM) 101 operating with a plurality of partitions 103-1 through 103-N, with each partition including workload manager (WLM) 10 and operating system (OS) 16. Ex. 1001, 2:52–54, 4:59–66. Although not shown in Figure 1, each partition also includes, *inter alia*, CPUs (processors), applications, a performance monitor, and a goals configuration file containing application goals and priorities. *Id.* at Fig. 2, 6:6–12, 6:30–38. PLM 101 receives resource requests from the partitions, compares the requests with available resources, and allocates resources based on performance goals and priorities, either within a partition among applications, or between partitions. *Id.* at 3:25–31, 3:38–44, 4:21–29,

5:14–16. Although Figure 1 depicts the PLM as separate from the partitions, it may reside in one partition and have access to the other partitions, or reside in each partition, and cooperatively allocate resources amongst themselves. *Id.* at 4:14–19.

Each WLM receives goal and priority information from the goals configuration file (provided by a user or administrator), and receives performance information from the performance monitor, which monitors the performance of the applications and devices within the partition. Ex. 1001, 3:58–59, 6:30–38. The WLM maintains a list of prioritized application goals with an indication of the quantity of each required resource. *Id.* at 4:24–27. The WLM compares the monitored information with the goals and priorities, and based on the comparison sends resource requests to the PLM. *Id.* at 3:65–4:9, 5:23–25, 6:30–38. The PLM receives the requests, and “allocates the resources between the different partitions, based on the priorities of the partitions and the resource requests.” *Id.* at 4:21–23, 7:13–17. “WLM, after receiving information from PLM about its partition resources, then allocates its allotted resources among the applications on its partition.” *Id.* at 4:11–13.

Figure 7, reproduced below, illustrates a virtual machine embodiment of the '005 patent subject matter.

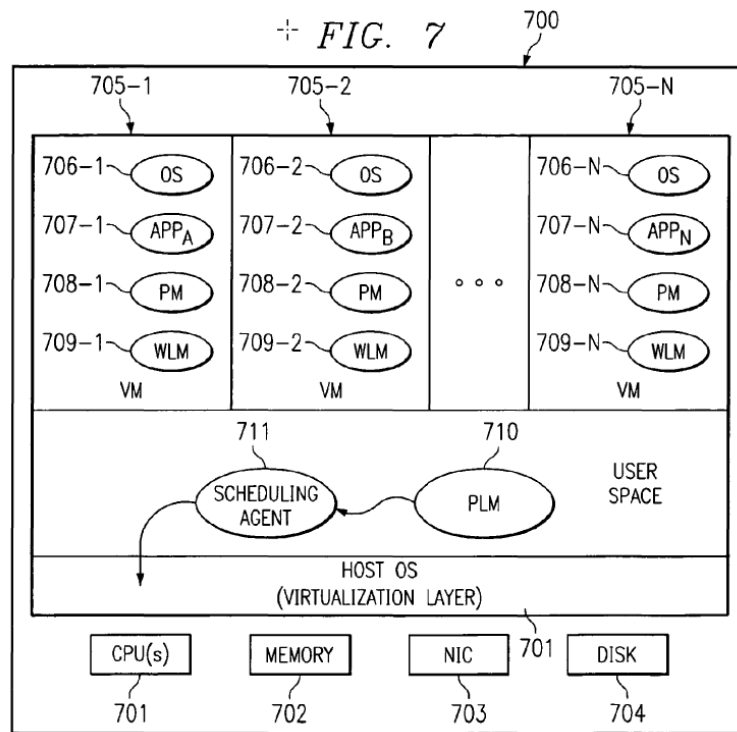


Figure 7 depicts a system that allocates virtual resources, based on underlying physical resources including CPUs 701, memory 702, network interface card (NIC) 703, and disk storage 704, and a host operating system that enables low level access to the physical resources. Ex. 1001, 13:62–67. The host operating system includes a virtualization software layer that virtualizes the physical resources as software constructs, creating virtual machines 705-1 through 705-N. *Id.* at 13:45–48, 13:65–14:5. The virtual machines “appear to software processes executing within the virtual machines to be physical server platforms,” and “provide partition and isolation functionality” such that a “software fault within any particular virtual machine 705 may only affect the respective virtual machine 705, while software processes associated with the other virtual machines 705 may continue operations in an ordinary manner.” *Id.* at 14:5–13.

Each virtual machine includes guest operating system (OS) 706, applications (APP) 707, performance monitor (PM) 708, and workload manager (WLM) 709. Ex. 1001, 14:13–23. Partition load manager (PLM) 710 and scheduling agent 711 are software processes that operate in the user space associated with the host operating system. *Id.* at 14:27–37. The functional interaction between WLMs and PLM is essentially the same as that described above for the physical partition embodiment: WLM receives performance data related to applications from the performance monitor and compares that to goals and priorities (also referred to as “SLOs,” or service level objectives), and when an application is not achieving its goals, WLM requests additional resources from PLM, which reallocates resources as appropriate. *Id.* at 14:21–34. PLM communicates with scheduling agent 711 which, for example, reassigns time slices of processors between virtual machines. *Id.* at 14:37–40.

B. The Prosecution History of the '005 Application

During prosecution of the '005 application, the Examiner rejected independent application claim 8 as being obvious over Eilert (Ex. 1007). Ex. 1002, 143–145. Applicants overcame the rejection by amending application claim 8 to add the requirement that the “manager process maintains a list comprising a plurality of application priority levels for said at least one application and an indication of a quantity of said plurality of resources needed to meet said at least one level of priority.” *Id.* at 89. Ultimately, the claim 8 was allowed with this amendment and additional Examiner’s amendments. *Id.* at 14–25.

C. Illustrative Claim

Independent claim 8 is reproduced below (with Petitioner's identifiers of limitations in brackets).

8. A method, comprising:

[8.1] creating a plurality of computing domains;

[8.2] allocating a plurality of resources between said plurality of computing domains by a first manager process;

[8.3] executing at least one application, a second manager process, and a performance monitor process in each of said plurality of computing domains,

[8.4] wherein said second manager process maintains a list comprising a plurality of application priority levels for said at least one application and an indication of a quantity of said plurality of resources needed to meet said at least one level of said application priority levels and

[8.5] wherein said performance monitor process generates performance data related to said at least one application and

[8.6] said second manager process requests additional resources from the first manager process in response to analysis of said performance data in view of at least one service level parameter; and

[8.7] dynamically reallocating said plurality of resources between said plurality of computing domains by the first manager process in response

to received requests for additional resources
according to service level parameters.

Ex. 1001, 15:46–16:3; Pet. ix.

D. Asserted Challenges to Patentability and Evidence of Record

Petitioner challenges the patentability of claims 8–14 of the '005 patent based on the following references:

Claims Challenged	35 U.S.C. §	Reference(s)/Basis
8	103 ¹	Eilert, ² Culbert ³
9–14	103	Eilert, Culbert, Bugnion ⁴
8	103	Boland, ⁵ Culbert
8–14	103	Boland, Culbert, Bugnion

Pet. 4. In support of its patentability challenge, Petitioner relies on, *inter alia*, the Declaration of Erez Zadok, Ph.D. Ex. 1003 (“Zadok Decl.”).

E. Real Parties in Interest

Petitioner identifies itself as a real party in interest. Pet. 2.

Patent Owner states:

Valtrus Innovations Limited (“Valtrus”) is the owner of U.S. Patent No. 7,748,005 (“the '005 patent”). Valtrus Innovations Limited is a subsidiary of Key Patent Innovations Limited.

¹ The Leahy-Smith America Invents Act, Pub. L. No. 112-29, 125 Stat. 284 (2011) (“AIA”), included revisions to 35 U.S.C. § 103 that became effective as of March 16, 2013. The application for the '005 patent was filed before March 16, 2013, and therefore we apply the pre-AIA version of 35 U.S.C. § 103.

² Eilert et al., US 6,587,938 B1 (“Eilert,” Ex. 1007).

³ Culbert, US 5,838,968 (“Culbert,” Ex. 1005).

⁴ Bugnion et al., US 6,075,938 (“Bugnion,” Ex. 1006).

⁵ Boland, US 6,947,987 B2 (“Boland,” Ex. 1004).

Both Key Patent Innovations Limited and Valinvest LLC are financially interested in the outcome of the litigation between Valtrus and Google.

Paper 4, 1.

F. Related Proceedings

The parties identify the following related proceedings: *Valtrus Innovations Ltd. v. Google LLC*, Case No. 3:22-cv-00066 (N.D. Tex.); *Valtrus Innovations Ltd. v. Google LLC*, Case No. 4:22-cv-00020 (N.D. Tex.). Pet. 2; Paper 4, 1–2.

In addition, *Google LLC v. Valtrus Innovations Limited*, IPR2022-01406 and *Google LLC v. Valtrus Innovations Limited*, IPR2022-01408 challenge claims of the '005 patent. Prelim. Resp. 2.

III. ANALYSIS OF PETITIONER'S CHALLENGE

A. Legal Standards

To prevail in its challenge, Petitioner must demonstrate by a preponderance of the evidence that the claims are unpatentable. 35 U.S.C. § 316(e); 37 C.F.R. § 42.1(d). “In an IPR, 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) (2012) (requiring *inter partes* review petitions to identify “with particularity . . . the evidence that supports the grounds for the challenge to each claim”). This burden of persuasion never shifts to the patent owner. *See Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (discussing the burden of proof in *inter partes* review).

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 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 ordinary skill in the art; and (4) when in evidence, objective indicia of non-obviousness (also called secondary considerations), such as commercial success, long-felt but unsolved needs, and failure of others.⁶ *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). We analyze grounds based on obviousness in accordance with the above-stated principles.

B. Level of Ordinary Skill in the Art

In determining whether an invention would have been obvious at the time it was made, 35 U.S.C. § 103(a) requires us to resolve the level of ordinary skill in the pertinent art at the time of the invention. *Graham*, 383 U.S. at 17. The person of ordinary skill in the art is a hypothetical person who is presumed to have known the relevant art at the time of the invention. *In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995). Factors that may be considered in determining the level of ordinary skill in the art include, but are not limited to, the types of problems encountered in the art, the

⁶ No evidence of secondary consideration appears in the record for us to consider at this stage of the proceeding.

sophistication of the technology, and educational level of active workers in the field. *Id.* In a given case, one or more factors may predominate. *Id.*

Petitioner asserts that a person of ordinary skill in the art:

would have had at least a bachelor's degree in computer science, computer engineering, or a related field, and at least two years of practical experience in the management of resources in a computing system. . . . These descriptions are approximate, and a higher level of education or skill might make up for less experience, and vice-versa.

Pet. 10 (citing Zadok Decl. ¶ 104). Patent Owner does not specifically address the level of ordinary skill in the pertinent art at the time of the invention. *See generally* Prelim. Resp.

Petitioner's proposal is consistent with the level of ordinary skill in the art reflected by the asserted prior art. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001); *GPAC*, 57 F.3d at 1579. On this record, the level of ordinary skill is not dispositive of any challenge. For purposes of this Decision, we apply Petitioner's articulation.

C. Claim Construction

The Petition was accorded a filing date of August 10, 2022. Paper 5, 1. In an *inter partes* review for a petition filed on or after November 13, 2018, a claim "shall be construed using the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b)." 37 C.F.R. § 42.100(b). We apply the claim construction standard from *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc).

Claim terms need only be construed to the extent necessary to resolve the controversy. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co. Matal*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

Neither Petitioner nor Patent Owner proposes claim constructions beyond asserting that the claim terms should be given their plain and ordinary meaning. Pet. 10; Prelim. Resp. 16. For purposes of this Decision, we do not construe any claim terms.

D. Alleged Obviousness of Claim 8 Over Eilert and Culbert

Petitioner challenges claim 8 as obvious over the combination of Eilert and Culbert. Pet. 10–34.

1. Eilert

Eilert, titled “Method, System And Program Products For Managing Central Processing Unit Resources Of A Computing Environment,” issued July 1, 2003 from a September 28, 1999 application. Ex. 1007, codes (54), (45), (22). Eilert is directed to dynamic reallocation of CPU resources among multiple logical partitions. *Id.* at Abstr., 2:25–37. Figure 1A is reproduced below.

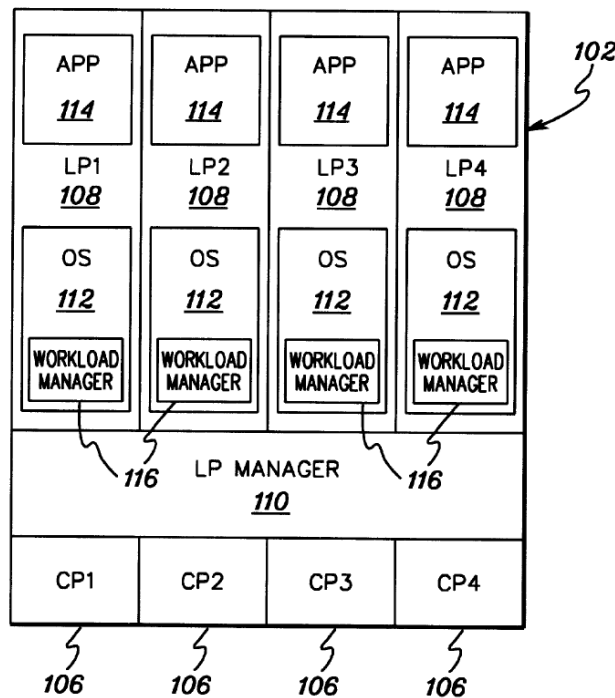


fig. 1A

Figure 1A depicts an exemplary computer environment, referred to as central processor complex (CPC) 102, based on International Business Machines Corporation's "Enterprise Systems Architecture (ESA)/390," which includes physical central processors 106 (CP1–CP4), LP manager (LPM) 110, and logical partitions 108 (LP1–LP4). Ex. 1007, 3:1–3, 4:1–17. Each logical partition has a logical processor (not shown), to which is allocated a portion of the physical CP resources, and on which run workload manager 116 (WLM), operating system 112 (OS), and applications 114 (APP). *Id.* at 4:19–38.

Logical partitions LP are managed by LPM 110, an example of which is IBM's Processor Resource/Systems Manager (PR/SM). *Id.* at 2:1–7, 4:50–56. According to the PR/SM manual, incorporated by reference in Eilert, LPM 110 enables the logical partitioning function of CPC 102, which

involves an operator defining the resources that are to be allocated to each logical partition. *Id.* at 2:1–7; Ex. 1011, 29.

Workload managers manage the workload within a partition and among partitions, whereby workload managers dynamically redistribute physical resources across the logical partitions. Ex. 1007, 4:37–38, 5:61–64, 8:49–50. Such resources include CPU resources, logical processor resources, I/O resources, coprocessors, channel resources, network adapters, and memory resources. *Id.* at 5:66–6:2. For example, the workload manager can reallocate CPU resources among logical partitions by varying the relative CPU processing priority weight that each logical partition is assigned. *Id.* at 7:10–23, 10:40–52. To enable the reallocation process, the workload managers have access to a shared data structure containing performance data about each logical partition, including:

CPU requirements of work running on the partition by service class; [h]ow well each service class is doing towards its goal on the partition;[and] CPU usage by CPU dispatching priority for the partition.

Ex. 1007, 12:8–18 (formatting changed). Thus, the workload manager reallocates resources among partitions “based on monitoring activity and customer’s goals.” *Id.* at 25:56–62.

2. *Culbert*

Culbert, titled “System And Method For Dynamic Resource Management Across Tasks In Real-Time Operating Systems,” issued November 17, 1998 from a March 1, 1996 application. Ex. 1005, codes (54), (45), (22). Culbert is directed to dynamic resource management across tasks in real-time operating systems. *Id.* at Abstr., 1:16–18. An illustrative example involves a multitasking environment made up of a host

system, a media engine subsystem, and a multimedia I/O subsystem. *Id.* at Fig. 1, 5:22–30. The host system includes a resource manager that dynamically manages system resources (*e.g.*, processor, bus and memory utilization) among the various tasks in the multitasking environment. *Id.* at 3:46–48, 5:31–36, 6:51–54, 7:2–6.

Each task in the multitasking environment is responsible for deciding the amount of each resource it requires, and to that end each task has an associated task resource utilization vector, illustrated in Figure 3 reproduced below. Ex. 1005, Fig. 3, 7:41–44.

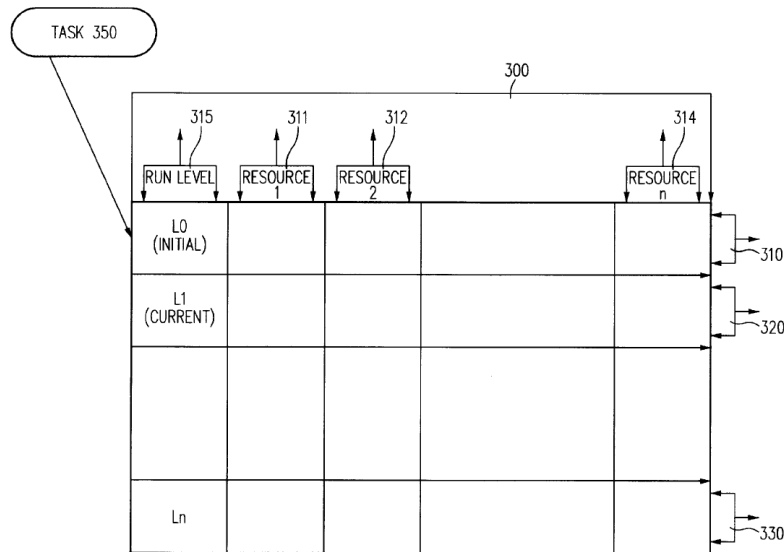


FIG. 3

Figure 3 illustrates task resource utilization vector 300 associated with task 350, made up of task resource utilization records 310, 320, through 330. *Id.* at 7:42–51. Task resource utilization vectors can be stored, *inter alia*, in the task or in the resource manager. *Id.* at 61–65. Each task resource utilization record includes run level index 315 and resource entries 311 through 314, where each resource entry contains the quantity of a resource requested for that run level. *Id.* at 7:51–55. The task resource utilization records are

sorted in preferred order, based on the favorability of the resource allocation to the task. *Id.* at 8:1–18. Run levels can be indicated symbolically, *e.g.*, optimal, next, and minimal; or indicated numerically, *e.g.*, 1, 2, . . . n. *Id.* at 8:40–41. When a task is running at a particular run level, the corresponding resource entries are periodically updated with the actual current resource usage quantities, using an “UpdateResourceMeasurement” routine, activated by a timer on a periodic basis. *Id.* at Table 1, 8:43–9:15.

The resource manager maintains a task list sorted in priority order, first to last, in the form of a list of vectors, each vector containing the tasks of a particular priority level, and each task containing its priority value and its task resource vector. Ex. 1005, Figs. 4, 4A, 9:57–10:9. Reallocation of resources is triggered by events such as a new task requiring more resources that are currently available, or a currently running task needing more resources, in which event the task manager carries out a “degradation process.” *Id.* at 9:16–23, 9:41–45. The resource manager seeks to obtain sufficient resources by degrading tasks in reverse order of priority, where degraded tasks free up resources by falling back to a different run level that uses less resources. *Id.* at 9:24–36, 11:10–19. The task manager can also trigger a similar “promotion” process triggering an upgrade in run level, if, for example, a client (*i.e.*, a user or administrator) requests a task promotion, or resources become available due to task termination or a task freeing resources. *Id.* at 11:53–65.

3. *Claim 8*

a) Petitioner's Arguments

In challenging claim 8, Petitioner argues:

A POSITA would have been motivated to use Culbert's teachings of centrally managing resource allocation via a resource manager, tracking application resource needs via a task resource allocation vector, updating that vector periodically with actual resource usage, and task demotion/promotion routines, in Eilert's resource distribution method.

Pet. 14 (citing Zadok Decl. ¶ 148). In particular, Petitioner argues that Eilert's logical partition manager 110 would have been modified to centrally manage resources, including handling requests for resources from workload manager 116, that workload manager 116 would have been modified to track resource needs and priorities, and that a performance monitor would have been in the same partition as the workload manager and would have been modified to update a resource vector with actual usage. *Id.*; *see also* Zadok Decl. ¶ 149. In support, Petitioner's declarant Dr. Zadok testifies that centrally managing resource allocation via a resource manager was well known. Zadok Decl. ¶¶ 73–74, 150.

Petitioner argues the combination would have been motivated given Culbert's statement that dynamic resource allocation “improve[s] overall system utilization and optimization” through use of the periodic updating of the task resource allocation vectors and demotion/promotion routines. Pet. 14–15 (citing Ex. 1005, Figs. 5–6, 4:21–24, 8:24–12:3; Zadok Decl. ¶¶ 151–152). Dr. Zadock testifies that one of ordinary skill would have recognized that these aspects of Culbert would have been compatible with Eilert and would have resulted in similar improvements, thus motivating the combination. Zadok Decl. ¶ 152.

Petitioner further argues that one of ordinary skill would have looked to Culbert for implementation details not present in Eilert, including using Culbert's task resource allocation vector to track resource needs and monitor activity, and Culbert's use of a central scheduler to integrate Eilert's workload manager and logical partition manager. Pet. 15–16 (citing Ex. 1007, 4:50–56, 10:50–52, 12:8–23, 23:42–44, 25:60–65; Zadok Decl. ¶¶ 155–156).

In overall support of the combination, Petitioner argues that both references are in the same field, achieving dynamic reallocation of system resources, that applying Culbert's teachings to Eilert apply known techniques to a known device, and that the combination uses known methods to combine known prior art elements. Pet. 15–17 (citing Ex. 1005, 3:15–4:26; Ex. 1007, 2:52–55; Zadok Decl. ¶¶ 153–154, 157–160). Dr. Zadok also asserts that the references are analogous art to the '005 patent, given the common goals of dynamically allocating system resources. Zadok Decl. ¶ 154 (citing Ex. 1001, 1:19–21, 3:6–7, 14:57–60; Ex. 1005, 3:15–4:29; Ex. 1007, 2:52–55).

Petitioner thus argues that the requirements of independent claim 8 would have been taught or suggested by this combination of Eilert and Culbert. Pet. 17–34. In particular, for the method preamble of claim 8, Petitioner generally relies on the methods for dynamically allocating resources disclosed in both references. Pet. 17 (citing Ex. 1007, Abstr.; Ex. 1005, Abstr.).⁷

⁷ Based on the present record, we make no determination at this stage of the proceeding that the preamble of claim 8 (or of claim 1) is limiting.

For requirement 8.1 of claim 8 regarding creation of computer domains,⁸ Petitioner relies on the logical partitions 108 (LP1–LP4) of Eilert. Pet. 17–19 (citing Ex. 1007, Fig. 1A, 3:60–65, 4:12–17; Zadok Decl. ¶¶ 169–170). As stated above in Section III.D.1, the IBM PR/SM, which is the example of the LPM 110 in Eilert, creates the logical partitions of CPC 102. Zadok Decl. ¶ 169. As stated in the '005 patent, logical partitions are examples of computer domains. Zadok Decl. ¶ 171 (citing Ex. 1001, 3:11–13, 13:36–41).

For requirement 8.2 of claim 8 regarding allocating resources between computing domains by a first manager process, Petitioner relies on the fact that Eilert allocates central processors 106 to logical partitions 108, and asserts that LPM 110 in Eilert, an example of which is the IBM PR/SM, is the required “first manager process” that performs that allocation. Pet. 19–20 (citing Ex. 1007, Fig. 1A, 3:60–65, 4:18–27, 4:50–56; Zadok Decl. ¶¶ 174–175). Dr. Zadok testifies that “[t]o the extent it is argued that Eilert is not explicit as to what does the allocation of resources between partitions, Eilert at least implicitly discloses that the logical partition manager 110 performs the allocation.” Zadok Decl. ¶ 175. Dr. Zadok cites the statements in Eilert that logical partitions 108 are managed by LPM 110, that resources are managed by LPM 110 when contention for resources occurs, and that dynamic adjustment of physical resources occurs “via, for instance, WLM, sysplex and PR/SM integration.” *Id.* (citing Ex. 1007, 4:50–56, 7:28–30, 25:60–65). Dr. Zadok cites the fact that the IBM PR/SM allocates

⁸ Following Petitioner’s approach, for economy of presentation we refer to the bracketed identifiers applied to the claim limitations by Petitioner (*see* Section II.C above).

processors and other resources to the different partitions that it creates. *Id.* ¶ 177 (citing Ex. 1011, 29, 41). Dr. Zadok additionally testifies that it would have been obvious for LPM 110 to perform the allocation because it “is logically between the physical system resources (e.g., CP 1–CP4) and the individual logical partitions (LP 1–LP4).” *Id.* ¶ 178.

Further to requirement 8.2, Petitioner also argues that Culbert teaches that “[r]esource manager 170 controls resource allocation and distribution for all XOS 180 tasks.” Pet. 21–22 (citing Ex. 1005, 6:51–54; Zadok Decl. ¶¶ 179–180). Dr. Zadok testifies that one of ordinary skill “would have recognized that resource manager 170 in Culbert is analogous to logical partition manager 110 in Eilert, and tasks in Culbert are analogous to the applications of Eilert.” Zadok Decl. ¶ 179.

For requirement 8.3 of claim 8 specifying that each domain contains executing applications, a second manager process, and a performance monitor, Petitioner relies on the disclosure that each logical partition 108 in Eilert includes execution of applications 114, and a workload manager 116. Pet. 22–23 (citing Ex. 1007, Fig. 1A, 3:60–65, 4:29–39, 4:63–5:8, 10:40–52, 25:56–65; Zadok Decl. ¶¶ 183–186). Petitioner asserts workload manager 116 is the second manager process, given that Eilert discloses that it manages the workload within a partition and among partitions, and that “physical resources are dynamically redistributed across the logical partitions . . . under direction of one or more workload managers.” Ex. 1007, 4:35–38, 5:61–64. For the performance monitor requirement, Petitioner relies on the statement in Eilert that shareable resources are provided to the needed partitions based on “monitoring activity and customers’ goals,” and that “each partition has access to a shared data

structure containing performance data.” *Id.* at 23 (citing Ex. 1007, 12:8–18, 25:57–62; Zadok Decl. ¶¶ 187–188). Dr. Zadok testifies that it would have been obvious that the monitoring activity, and collection of performance data, would have been performed by a process in each logical partition because the performance data would vary for each logical partition. Zadok Decl. ¶ 189.

For requirement 8.4 of claim 8 regarding the second manager process maintaining a list of priority levels and needed resources, Petitioner argues that a person of ordinary skill would have looked to Culbert’s teaching of resource allocation techniques for dynamically allocating resources, including Culbert’s use of task resource utilization records that include run levels, which Petitioner equates with priorities, and that also include required quantities of resources. Pet. 25–26 (citing Ex. 1005, Fig. 3, 7:41–60, 7:66–8:18, 8:38–46; Zadok Decl. ¶¶ 191–192). Dr. Zadok testifies that one of ordinary skill would have adapted Eilert to have each workload manager 116 maintain a list using the task resource utilization record approach of Culbert. Zadok Decl. ¶¶ 193–196.

For requirement 8.5 of claim 8 regarding performance monitoring, Petitioner relies on above described performance monitoring in Eilert, and also on the disclosure in Culbert that the resource usage data in the task resource utilization records is periodically updated using the UpdateResourceMeasurement process. Pet. 27–29 (citing Ex. 1005, 8:38–9:14; Ex. 1007, 12:8–18, 25:57–62; Zadok Decl. ¶¶ 198–203). Dr. Zadok testifies that a person of ordinary skill in the art would have recognized that the performance monitoring taught by Culbert would also be

applicable to the system of Eilert, including Culbert's use of the task resource utilization records. Zadok Decl. ¶¶ 200–203.

For requirement 8.6 of claim 8 regarding the second manager process requesting additional resources, Petitioner relies on Eilert's disclosure that the workload managers dynamically redistribute resources across the logical partitions based on monitoring activity and customers' goals, as teaching the requirement that the "second manager process requests additional resources . . . in response to analysis of said performance data in view of at least one service level parameter. Pet. 29–30 (citing Ex. 1007, 2:33–37, 25:57–62; Zadok Decl. ¶¶ 205–209). For the requirement that the second manager process requests additional resources *from the first manager process*, Petitioner relies on a statement in Eilert, "dynamic adjustment of physical resources in accordance with workload goals via, for instance, WLM, sysplex and PR/SM integration, is performed without needing parallel sysplex data sharing." Pet. 30 (citing Ex. 1007, 25:63–65; Zadok Decl. ¶ 210). (As discussed, "PR/SM" is an example of a logical partition manager, which Petitioner equates with the claimed first manager process). In addition, Petitioner relies on the disclosure in Culbert of the task promotion routine performed by resource manager 170, which Petitioner argues is analogous to the logical partition manager of Eilert. Pet. 31 (citing Ex. 1005, Fig. 6, 11:56–59, 12:14–22; Zadok Decl. ¶¶ 212–215). Dr. Zadok testifies that one of ordinary skill would have adapted the task promotion routine of Culbert into Eilert by having the workload manager request the logical partition manager to promote a task. Zadok Decl. ¶ 213.

For requirement 8.7 of claim 8 regarding dynamic reallocation of resources by the first manager process, Petitioner relies on the above

combination of Eilert and Culbert, in which the workload manager uses the task promotion routing of Culbert to promote a task, with the request directed to the logical partition manager. Pet. 33–34 (citing Ex. 1005, 3:20–26, 11:24–31, 11:52–13:3; Ex. 1007, 2:33–37, 25:57–62; Zadok Decl. ¶¶ 217–219).

b) Patent Owner’s Preliminary Response

Patent Owner argues that Petitioner provides insufficient reasons why a person of ordinary skill in the art would have been motivated to combine Eilert with Culbert in the manner asserted. Prelim. Resp. 58–61. Patent Owner argues that although both references relate to dynamic resource allocation, Petitioner does not explain how Culbert would improve resource allocation in Eilert. *Id.* at 58–59. Patent Owner submits that Petitioner’s arguments that both references are in the same field, and have analogous components, are not sufficient motivation to combine them. *Id.* at 59. Patent Owner challenges Petitioner’s assertion that Culbert provides “implementation details” missing from Eilert, arguing that Culbert does not add relevant details insofar as any claim requirements are concerned. *Id.* at 59–60. In addition, Patent Owner argues that Petitioner’s assertions regarding applying known techniques and using known methods to combine known prior art elements is a conclusory and unsupported attempt to invoke general legal principals inapplicable to the combination asserted here. *Id.* at 60–61.

Turning to specific requirements of claim 8, Patent Owner argues that the combination does not teach the required interaction between a first manager process and a second manager process — *i.e.*, requirement 8.6 that the second manager process requests additional resources from the first

manager process, and corresponding requirement 8.7 that the first manager process dynamically reallocates resources in response to the received requests. Prelim. Resp. 26–31, 46–52. Patent Owner argues that Eilert’s LPM 110 does not dynamically reallocate resources, but rather manages logical partitions in accord with static pre-defined allocations set up by the operator of the system. *Id.* at 27. Patent Owner explains that the statement in Eilert that Petitioner relies on, that LPM 110 manages resources according to defined container sizes when partitions contend for resources, is based on this static resource allocation and is not in response to any request from workload manager 116, which is Petitioner’s candidate for the second manager process. *Id.* at 27–28 (citing Ex. 1007, 7:28–31). Patent Owner cites the IBM PR/SM document, which describes an example of Eilert’s LPM 110 and which is incorporated by reference in Eilert, which states that the logical partition manager statically assigns resources to logical partitions based on operator pre-assigned processing weights. *Id.* at 28 (citing Ex. 1011, 29, 172, 296). Patent Owner emphasizes that “while LPM 110 creates and groups partitions, it plays no role in managing the workload and resources within and among the partitions.” *Id.* at 29.

In contrast to Eilert’s LPM 110, Patent Owner argues that Eilert’s workload manager 116 has the sole responsibility for reallocating resources, and does not request resources from LPM 110. Prelim. Resp. 29–31. As cited by Patent Owner, Eilert states that “workload manager can distribute CPU resources across logical partitions,” “each operating system . . . includes a workload manager 116 for managing the workload within a partition and among partitions,” and “physical resources are dynamically redistributed across the logical partitions . . . under direction of one or more

workload managers.” *Id.* at 29 (citing Ex. 1007, 2:44–46, 4:35–37, 5:61–64). Patent Owner argues that the reference in Eilert to “integration” between “WLM, sysplex, and PR/SM,” on which Petitioner and its declarant rely, does not support any inference that workload manager 116 requests resource allocations from LPM 110. *Id.* at 30 (citing Ex. 1007, 25:56–65). Indeed, argues Patent Owner, that portion of Eilert confirms that “shareable resources are dynamically redistributed across the [logical partitions] under workload manager direction.” *Id.* at 31 (citing Ex. 1007, 25:58–59).

Patent Owner further argues that nothing in Culbert justifies Petitioner’s and its declarant’s asserted combination of Eilert and Culbert that would modify LPM 110 and workload manger 116 to create the required request relationship. Prelim. Resp. 46–52. Patent Owner points out that Petitioner redundantly assigns Culbert’s resource manager 170 the role of the claimed first manager process, but does not explain how Eilert’s LPM 110 (also the asserted first manager) would be modified other than characterizing Culbert’s workload manager 116 as “centrally managing resources.” *Id.* at 47. Patent Owner argues that Petitioner’s assertion that Culbert’s resource manager 170 is “analogous” to Eilert’s LPM 110 is not reason why a skilled artisan would have made any pertinent modifications to LPM 110. *Id.* at 48. Patent Owner argues that “Ppetitioner has provided no reason a skilled artisan would have ever turned to Culbert to modify Eilert’s LPM 110 . . . and abandon [its] existing allocation mechanisms,” or explained how LPM 110 would be reconfigured to implement Culbert’s resource manager 170 functionality. *Id.*

Patent Owner likewise argues that Petitioner does not justify incorporating the task promotion routine of Culbert into workload manager

116 of Eilert. Prelim. Resp. 49–52. Patent Owner argues that Eilert’s lack of explanation of what it means for LPM 110 and workload manager 116 to be “integrated” is not resolved by incorporating Culbert’s task promotion routine, and further argues that splitting the allocation of resources between Eilert’s workload manager 116 and LPM 110 is the opposite approach of both Eilert and Culbert, which both use a single manager to reallocate resources. *Id.* at 49–50. Patent Owner argues that Petitioner does not explain why incorporating the task promotion routine of Culbert into Eilert would cause workload manager 116 to generate resource requests, when in Culbert it is the tasks (which Petitioner equates to the claimed applications) that request promotion. *Id.* at 50–51.

Turning to requirement 8.3 of claim 8, which requires a performance monitor process in each domain, and requirement 8.5, which requires the performance monitor to generate performance data for an application, Patent Owner argues that Eilert does not disclose a specific performance monitor process, or that there any such process in each logical partition. Prelim. Resp. 32. Instead, argues Patent Owner, the performance data in Eilert is located in a centralized “shared data structure,” and the process monitors external resource usage by partitions, not internal application performance. *Id.* at 32–33 (citing Ex. 1007, 10:45–52, 12:8–18). Patent Owner criticizes as conclusory Petitioner’s explanation that, because performance data would vary for each partition, each partition would have a performance monitor. *Id.* at 33. Patent Owner also argues that the monitoring in Eilert does not satisfy the requirement of generating performance data “related to at least one application,” because the activity in a partition is the result of multiple components, and not necessarily correlated to an application. *Id.* at 34–35.

Patent Owner further argues that Petitioner does not sufficiently support incorporating Culbert's updating of a task resource utilization vector into the performance monitoring process of Eilert. Prelim. Resp. 56–57. Patent Owner argues that Eilert already describes monitoring sufficient to its system, and there is no demonstrated need for further modification. *Id.* at 57.

Finally, for requirement 8.4 of claim 8, requiring the second manager process to maintain a list of application priority levels and resources needed to meet the priority levels (which Patent Owner characterizes as “key to distinguishing Eilert during prosecution”), Patent Owner argues that Petitioner provides no reason or evidence for how or why a skilled artisan would have been motivated to combine Culbert's task resource utilization vector with Eilert's workload manager in the claimed manner. Prelim. Resp. 53. Patent Owner argues that Petitioner's conclusory assertions that one of ordinary skill could have adopted the Culbert task resource utilization vector in Eilert, or that Culbert satisfied unspecified “implementation details” lacking in Eilert, are an insufficient basis for the combination. *Id.* at 53–54. Patent Owner submits that Eilert's workload manager 116 distributes resources based on resource usage by partitions, and has no missing details that need to be remedied by Culbert. *Id.* at 54. Moreover, Patent Owner points out that the task resource utilization vector in Culbert resides in the task or in resource manager 170, which Petitioner equates to the '005 patent's applications and first manager process, respectively, and so there would be no reason to insert the task resource utilization vector into workload managers 116, which Patent Owner equates to the second manager process. *Id.* at 55–56 (citing Ex. 1005, 7:61–65).

c) Analysis

We agree with Patent Owner that Petitioner has not sufficiently shown that several claim 8 requirements are taught or suggested by the combination of Eilert and Culbert. In particular, we are not persuaded that Eilert and Culbert, taken alone or in combination, render obvious the required first and second manager processes, where the first manager process reallocates resources in response to requests from second manager processes in each computer domain. Culbert, which is a single multitasking system not made up of multiple domains, reallocates resources to tasks using a single manager process — resource manager 170. Ex. 1005, Fig. 1, 3:16–18, 5:31–36, 6:51–54. Likewise, Eilert’s workload manager 116, in each partition, reallocates resources based on monitoring activity and customer’s goals. Ex. 1007, 4:36–39, 5:61–64, 25:56–62. Although Eilert’s LPM 110 statically allocates resources to logical partitions based on operator pre-assigned processing weights, we are not persuaded that it has, or would be modified to have, a role in dynamically reallocating resources, or respond to requests for resources from workload managers 116. Ex. 1007, 4:50–52; Ex. 1011, 29, 172, 296. Petitioner characterizes the Culbert resource manager as “centrally control[ing] resource allocation” (Pet. 22), and argues that this would motivate adopting the LPM 110 of Eilert to such a central role, with workload managers 116 sending to it requests for resources. Pet. 22, 31. But the resource manager of Culbert serves a multitasking system that is not made up of multiple domains, and therefore the resource manager has no relationship with other managers that could be characterized as “central,” and thus would not have served as a motivation to fundamentally

change the distributed workload manager approach of Eilert. Nor do we see any other reason to fundamentally change the operation of Eilert.

In addition, the record does not support Petitioner's assertion that the combination of Eilert and Culbert teaches or suggest including a performance monitor in each logical partition, per requirement 8.3 of claim 8. In Eilert, the only disclosure of a performance monitor process states that performance data is in a "shared data structure," suggesting that the monitoring is external to the logical partitions. Ex. 1007, 12:8–12. Also, there is no indication in Culbert of the location of the "UpdateResourceMeasurement routine," which periodically updates the task resource utilization records, and therefore Culbert provides no basis for motivating a redesign of Eilert to insert performance monitors in each partition.

Finally, we are not persuaded that there is sufficient motivation to modify workload managers 116 to include Culbert's resource utilization records, which Petitioner equates to the list of application priority levels and resources needed to meet the priority levels as per requirement 8.4 of claim 8. In Culbert, the resource utilization records are included in the task structure itself, the resource manager 170, or media operating system 180. Ex. 1005, 7:61–63. Therefore, if anything, Eilert would have been modified to include the resource utilization records in the applications or LPM 110, which are the alleged counterparts to the Culbert tasks and resource manager. In addition, we do not understand the basis for Petitioner's argument that the run levels of the resource utilization records correspond to the required application priority levels, given that Culbert discloses an

entirely separate “task list,” with entries including the priorities of each task, and which has nothing to do with run levels. Ex. 1005, Fig. 4, 9:57–10:2.

Accordingly, for the reasons set forth above, we determine that the Petition does not demonstrate a reasonable likelihood that Petitioner would prevail in establishing that claim 8 would have been obvious over the combination of Eilert and Culbert.

E. Alleged Obviousness of Claims 9–14 Over Eilert, Culbert, and Bugnion

Petitioner challenges claims 9–14 as obvious over the combination of Eilert, Culbert, and Bugnion. Pet. 34–44. Petitioner’s challenge does not rely on Bugnion for any of the requirements of claim 8, discussed above, that underlie the dependent claims. *Id.*

Accordingly, given our determination that the Petition does not demonstrate a reasonable likelihood that Petitioner would prevail in establishing that independent claim 8 would have been obvious over the combination of Eilert and Culbert, we likewise determine that the Petition does not demonstrate a reasonable likelihood that Petitioner would prevail in establishing that dependent claims 9–14 would have been obvious over the combination of Eilert, Culbert, and Bugnion.

F. Alleged Obviousness of Claim 8 Over Boland and Culbert

Petitioner challenges claim 8 as obvious in light of the combination of Boland and Culbert. Pet. 45–66. The parties’ arguments, and our analysis, for the challenges based on the Boland/Culbert combination closely track those of the Eilert/Culbert combination.

1. Boland

Boland, titled “Method And Apparatus For Allocating Network Resources And Changing The Allocation Based On Dynamic Workload Changes,” issued September 20, 2005 from a May 29, 1998 application. Ex. 1004, codes (54), (45), (22). Boland is directed to allocating and distributing processes to network resources. *Id.* at Abstr. Figure 3 is reproduced below.

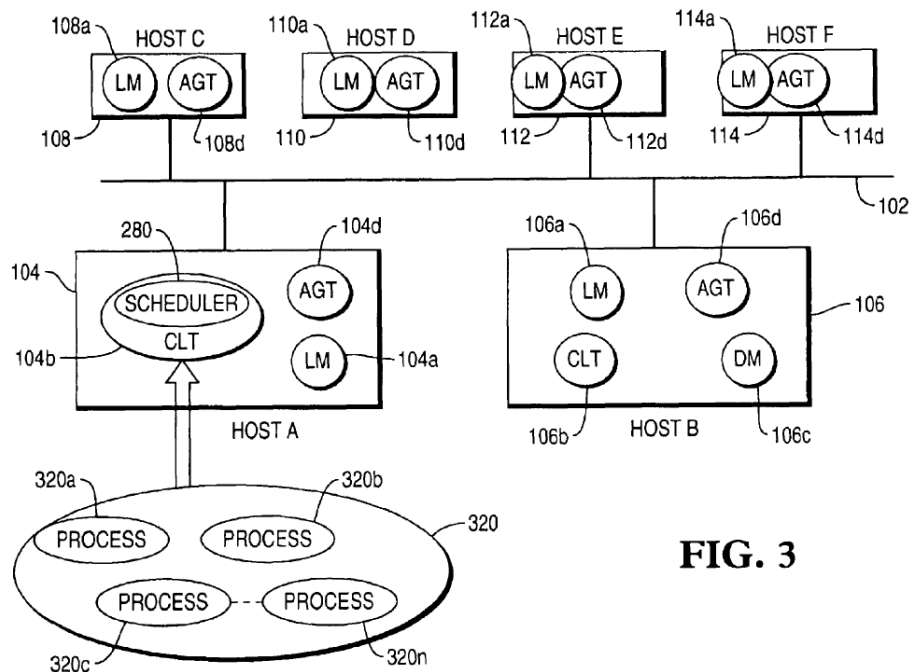


FIG. 3

Figure 3 is a logical view of a computer network made up of host computers A through E running processes 320a through 320n. Ex. 1004, 3:45–47, 5:35–44. Each host computer has a processor 204 (not shown). *Id.* at Fig. 2, 4:34–37. Host 104 includes scheduler 280, which is a process allocating device that allocates processes 320a, 320b, 320c ... 320n to be executed on computer systems 104, 106, 108, 110, 112, and 114. *Id.* at 5:44–49. Clients (CLT) 104b and 106b are associated with Hosts A and

B, which clients can request computers usable for process distribution. *Id.* at 5:52–57. Host B (106) includes a main domain resource manager (DM) 106c that holds the data on the operational state of each computer 104–114 and determines the usable computers when a client CLT requests computer resources. *Id.* at 5:58–63. Each computer system has a resource use permitting mechanism agent (AGT) 104d, 106d, 108d, 110d, 112d, 114d for supplying a message to domain resource manager DM (106c), which permits the exclusive use of a computer resource to client CLT by domain resource manager DM. *Id.* at 5:63–6:2. Each computer 104–114 includes a local resource manager (LMs) 104a, 106a, 108a, 110a, 112a, 114a residing in the respective computer processors to monitor their operational states and transmit, to domain resource manager (DM 106c), situation data which indicate the processor 204 usage of the computers. *Id.* at 5:49–52, 6:3–6.

Scheduler 280 is used for the distribution of processes on the network, based on execution record data, information on the amount of processing required by each of the processes, data parameters such as the processor speed and the processor load (usage), and rules for allocating processes including a minimum resource allocation and the priorities for processing allocation for each process, as set by a systems administrator. Ex. 1004, 6:30–54, 6:67–7:4, 7:13–15. Domain resource manager 106c, working in conjunction with scheduler 280, reallocates the network resources. *Id.* at 7:46–48. As stated in Boland:

The minimum resource allocation means that the process having the highest priority will be allocated resources before other processes. Each of the network resources is monitored for resource use. If necessary, a resource allocator redistributes network resources in accordance with the minimum resource allocation.

Ex. 1004, 2:33–39.

2. *Claim 8*

a) Petitioner's Arguments

In challenging claim 8, Petitioner argues:

A POSITA would have been motivated to use Culbert's teachings of centrally managing resource allocation via a resource manager . . . , tracking application resource needs via a task resource allocation vector and updating that vector periodically with actual resource usage, and task demotion/promotion routines, in Boland's resource distribution method.

Pet. 47 (citing Zadok Decl. ¶ 262). In particular, Petitioner argues that Boland's scheduler 280 would have been modified to centrally manage resources, including handling requests for task promotion received from an agent 104d, . . . , 112d, that the agents would have been modified to track resource needs for the applications they are managing, and local resource monitors would have been modified to update a Culbert-type resource vector with actual usage. *Id.*; *see also* Zadok Decl. ¶ 263. In support, Petitioner's declarant Dr. Zadok testifies that centrally managing resource allocation via a resource manager was well known. Zadok Decl. ¶¶ 73–74, 264.

Petitioner argues the combination would have been motivated given Culbert's statement that dynamic resource allocation “improve[s] overall system utilization and optimization” through use of the periodic updating of the task resource allocation vectors and demotion/promotion routines. Pet. 47–48 (citing Ex. 1005, Figs. 5–6, 4:21–24, 8:24–12:3; Zadok Decl. ¶ 265). Dr. Zadock testifies that one of ordinary skill would have recognized that these aspects of Culbert would have been compatible with Boland would

have resulted in similar improvements, thus motivating the combination. Zadok Decl. ¶ 266.

Petitioner further argues that one of ordinary skill would have looked to Culbert for implementation details not present in Boland, including using Culbert's task resource allocation vector to track resource needs and monitor activity, and Culbert's use of a central scheduler to provide details on the agents in Boland. Pet. 49 (citing Zadok Decl. ¶¶ 269–270).

In overall support of the combination, Petitioner argues that both references are in the same field, involve allocating resources efficiently to execute applications, that applying Culbert's teachings to Boland apply known techniques to a known device, and that the combination uses known methods to combine known prior art elements. Pet. 48–50 (citing Ex. 1005, 3:15–4:29; Ex. 1004, 2:2–7, 6:17–24, 7:61–8:3; Zadok Decl. ¶¶ 267, 271–274). Dr. Zadok also asserts that the references are analogous art to the '005 patent, given the common goals of dynamically allocating system resources. Zadok Decl. ¶ 268 (citing Ex. 1001, 1:19–21, 3:6–7, 14:57–60; Ex. 1005, 3:15–4:29; Ex. 1004, 2:2–8).

Petitioner thus argues that the requirements of independent claim 8 would have been taught or suggested by this combination of Boland and Culbert. Pet. 51–66. In particular, for the method preamble of claim 8, Petitioner generally relies on the methods for dynamically allocating resources disclosed in both references. Pet. 17 (citing Ex. 1004, 1:8–12; Ex. 1005, Abstr.).⁹

⁹ As stated above, we make no determination at this stage of the proceeding that the preamble of claim 8 (or of claim 1) is limiting.

For requirement 8.1 of claim 8 regarding creation of computer domains,¹⁰ Petitioner relies on the hosts A and B of Boland which include client (CLT) 104b and 106b that each “request computers usable for process distribution.” Pet. 51–52 (citing Ex. 1004, Fig. 3, 5:52–63, 6:26–54, 7:37–55; Zadok Decl. ¶¶ 283–287). Petitioner argues Host A and its assigned computers are one domain and Host B and its computers are another. *Id.* (citing Zadok Decl. ¶ 276). As stated in the ’005 patent, “[t]he computer system may comprise ... a network of larger and/or small computers” and that “the network may be divided into protection domains or partitions.” Zadok Decl. ¶ 286 (citing Ex. 1001, 3:8–13).

For requirement 8.2 of claim 8 regarding allocating resources between computing domains by a first manager process, Petitioner argues that Boland’ scheduler 280 is the required “first manager process” that allocates processes 320a, 320b, 320c, ..., 320n to be executed on the plurality of computer systems 104, 106, 108, 110, 112, 114, and thereby allocates processor resources 204, memory resources 206, and storage devices 210 in computer systems 108–114 (hosts C–F) to those processes. Pet. 53–54 (citing Ex. 1004, Figs. 2–3, 5:44–49, 6:11–16, 6:55–54, 7:45–47; Zadok Decl. ¶ 289). Petitioner also argues that Culbert teaches that “[r]esource manager 170 controls resource allocation and distribution for all XOS 180 tasks.” Pet. 55 (citing Ex. 1005, 6:51–54; Zadok Decl. ¶¶ 290–291). Dr. Zadok testifies that one of ordinary skill “would have recognized that resource manager 170 in Culbert is analogous to scheduler 280 in Boland,

¹⁰ Following Petitioner’s approach, for economy of presentation we refer to the bracketed identifiers applied to the claim limitations by Petitioner (*see* Section II.C above).

and tasks in Culbert are analogous to the processes of Boland.” Zadok Decl. ¶ 290.

For requirement 8.3 of claim 8 specifying that each domain contains executing applications, a second manager process, and a performance monitor, Petitioner relies on the disclosure that processes 320a – 320n “can be a third party application or program or a network process,” which correspond to the at least one application. Pet. 56 (citing Ex. 1004, 5:35–41; Zadok Decl. ¶ 294). Petitioner equates agents (AGT) 104d – 114d of each host to the second manager process. *Id.* (citing Ex. 1004, Fig. 3, 5:63–6:2; Zadok Decl. ¶¶ 278, 295–296). For the performance monitor requirement, Petitioner relies on “local resource manager (LMs) 104a, 106a, 108a, 110a, 112a, 114a” of each host “to monitor the operational states thereof,” and to “transmit situation data which indicate the processor 204 usage or the like of a respective computer to the domain resource manager (DM 106c).” *Id.* at 56–57 (citing Ex. 1004, Fig. 3, 2:36–37, 5:49–52, 6:3–11; Zadok Decl. ¶¶ 278, 297–300).

For requirement 8.4 of claim 8 regarding the second manager process maintaining a list of priority levels and needed resources, Petitioner argues that a person of ordinary skill would have looked to Culbert’s teaching of resource allocation techniques for dynamically allocating resources, including Culbert’s use of task resource utilization records that include run levels, which Petitioner equates with priorities, and that also include required quantities of resources. Pet. 58–60 (citing Ex. 1005, Fig. 3, 7:41–60, 7:66–8:18, 8:38–46; Zadok Decl. ¶¶ 305–307). Dr. Zadok testifies that one of ordinary skill would have understood that each host’s agent (the second manager process) would be well-suited for maintaining the list of

application level priorities and resources needed (i.e., the task resource utilization vector of Culbert) for the set of applications (tasks) that is assigned to the host. Zadok Decl. ¶ 304.

For requirement 8.5 of claim 8 regarding performance monitoring, Petitioner relies on above described performance monitoring in Boland, and also on the disclosure in Culbert that the resource usage data in the task resource utilization records is periodically updated using the UpdateResourceMeasurement process. Pet. 61–62 (citing Ex. 1005, 8:38–9:14; Ex. 1004, 6:3–10; Zadok Decl. ¶¶ 280, 309–312). Dr. Zadok testifies that a person of ordinary skill in the art would have recognized that the performance monitoring taught by Culbert would also be applicable to the system of Boland, including Culbert’s use of the task resource utilization records. Zadok Decl. ¶ 312.

For requirement 8.6 of claim 8 regarding the second manager process requesting additional resources, Petitioner relies on Boland’s disclosure “that the agent of each host (i.e., the second manager process) coordinates with the domain resource manager to permit the use of computer resources by a client, such as clients 104b and 106b of hosts A and B.” Pet. 62 (citing Ex. 1004, 5:63–6:2, 7:15–17, 7:37–55; Zadok Decl. ¶ 314). For the requirement that the second manager process requests additional resources *from the first manager process*, Petitioner relies on the task promotion process of Culbert, whereby a client can request more resources for a task, and argues that one of ordinary skill would have understood from those teachings that an agent in Boland that is running a process may request promotion of that process from the scheduler 280 (i.e., the first manager process, which is analogous to resource manager 170 in Culbert) based on

comparing its actual resource usage as measured with its run level requirements. Pet. 63–64 (citing Ex. 1005, 11:52–13:3; Zadok Decl. ¶¶ 316–318).

For requirement 8.7 of claim 8 regarding dynamic reallocation of resources by the first manager process, Petitioner relies on the above combination of Boland and Culbert, in which Boland’s agent uses the task promotion routing of Culbert to promote a task, with the request directed to scheduler 280. Pet. 64–66 (citing Ex. 1005, 3:20–26, 11:24–31, 11:52–13:3; Ex. 1004, 2:36–39, 3:55–57, 6:43–48; 6:55–64; 7:37–55; Zadok Decl. ¶¶ 321–322).

b) Patent Owner’s Preliminary Response

Patent Owner argues that Petitioner provides insufficient reasons why a person of ordinary skill in the art would have been motivated to combine Boland with Culbert in the manner asserted. Prelim. Resp. 58–61. Patent Owner argues that although both references relate to dynamic resource allocation, Petitioner does not explain how Culbert would improve resource allocation in Boland. *Id.* at 58–59. Patent Owner submits that Petitioner’s arguments that both references are in the same field, and have analogous components, are not sufficient motivation to combine them. *Id.* at 59. Patent Owner challenges Petitioner’s assertion that Culbert provides “implementation detail” missing from Boland, arguing that Culbert does not add relevant details insofar as any claim requirements are concerned. *Id.* at 59–60. In addition, Patent Owner argues that Petitioner’s assertions regarding applying known techniques and using known methods to combine known prior art elements is a conclusory and unsupported attempt to invoke

general legal principals inapplicable to the combination asserted here. *Id.* at 60–61.

Turning to specific requirements of claim 8, Patent Owner argues that the combination does not teach requirement 8.1 of claim 8 requiring creating a plurality of computer domains because the asserted domains include a host computer A or B of Boland and also include associated computer resources running processes on additional hosts. Prelim. Resp. 35–38. Patent Owner argues that this proposed reading would cause the alleged computing domains to continually change based on what resources on the network are associated with which processes during allocation and reallocation, and create overlapping resource among different hosts. *Id.* at 36–37 (citing Ex. 1004, 6:17–24, 7:45–55). According to Patent Owner, this is inconsistent with the property of computing domains that they be independent of and protected from each other. *Id.* at 38 (citing Ex. 1001, 3:11–13, 4:59–5:3, 14:7–13). Patent Owner also argues that Petitioner’s analysis on based on conclusory expert testimony and insufficient hindsight support. *Id.*

Patent Owner also argues that the combination does not teach the required interaction between a first manager process and a second manager process — *i.e.*, requirement 8.6 that the second manager process requests additional resources from the first manager process, and corresponding requirement 8.7 that the first manager process dynamically reallocates resources in response to the received requests. Prelim. Resp. 39–42, 46–52. Patent Owner argues that Boland’s scheduler 280 does not dynamically reallocate resources to computing domains as required, but rather allocates

processes to multiple hosts. *Id.* at 39–40 (citing Ex. 1004, 2:26–28, 5:44–49, 6:30–31, 6:43–45, 7:27–28).

Patent Owner also disputes Petitioner’s assertion that the agents of Boland are the claimed second manager process, because there is no disclosure in Boland that agents request resources from scheduler 280, the alleged first manager process — Boland’s agents are only described as “supplying a message to the domain resource manager DM (106c) which permits the exclusive use of a computer resource to the client CLT by the domain resource manager DM.” Prelim. Resp. 40–41 (citing Ex. 1004, 5:66–6:2). Patent Owner also argues that the agents do not supply messages “in response to analysis of said performance data,” per requirement 8.6 of claim 8, because the alleged performance monitors, the local resource managers of Boland, send performance data to Boland’s domain resource manager (“DM”), not the agents. *Id.* at 41–42 (citing Ex. 1004, Fig. 3, 5:58–63, 6:3–10).

Patent Owner further argues that nothing in Culbert justifies Petitioner’s and its declarant’s asserted combination of Boland and Culbert that would modify Boland’s scheduler 280 or agents to create the required request relationship. Prelim. Resp. 46–52. Patent Owner points out that Petitioner redundantly assigns Culbert’s resource manager 170 the role of the claimed first manager process, but does not explain how Boland’s scheduler 280 (also the asserted first manager) would be modified other than characterizing Culbert’s workload manager 116 as “centrally managing resources.” *Id.* at 47. Patent Owner argues that Petitioner’s assertion that Culbert’s resource manager 170 is “analogous” to Boland’s scheduler 280 is not reason why a skilled artisan would have combined the references. *Id.*

at 48. Patent Owner argues that “Petitioner has provided no reason a skilled artisan would have ever turned to Culbert to modify . . . Boland’s scheduler 280 and abandon [its] existing allocation mechanisms,” or explained how scheduler 280 would be reconfigured to implement Culbert’s resource manager 170 functionality. *Id.*

Patent Owner likewise argues that Petitioner does not justify incorporating the task promotion routine of Culbert into the agents of Boland. Prelim. Resp. 49–52. Patent Owner argues that there is no basis for reliance on Culbert’s task promotion routine when agents merely “permit” the use of resources, but do not “run” the process and send requests to scheduler 280.” *Id.* at 51. Patent Owner further argues that Petitioner has no explanation for cutting out the role of the domain resource manager in Host B — given it is that component that interfaces with scheduler 280, not the agents. *Id.* at 51–52. In addition, Patent Owner argues that nothing in Culbert’s task promotion routine would have led to having Boland’s agent receive performance data rather than the domain resource manager. *Id.* at 52.

Turning to requirement 8.5 of claim 8, which requires the performance monitor to generate performance data for an application, Patent Owner argues that local resource managers of Boland do not generate performance data “related to said at least one application,” but rather send processor usage data and hardware information to the domain resource manager. Prelim. Resp. 45 (citing Ex. 1004, 6:3–10). Patent Owner also argues that Petitioner’s hypothetical where only one application is running in a host is without merit because the activity in a host is the result of multiple components, and not necessarily correlated to an application. *Id.* at 45–46.

Patent Owner further argues that Petitioner does not sufficiently support incorporating Culbert's updating of a task resource utilization vector into the performance monitoring process of Boland. Prelim. Resp. 56–57. Patent Owner argues that Boland already describes monitoring sufficient to its system, and there is no demonstrated need for further modification. *Id.* at 57.

Finally, for requirement 8.4 of claim 8, requiring the second manager process to maintain a list of application priority levels and resources needed to meet the priority levels (which Patent Owner characterizes as “key to distinguishing Eilert during prosecution”), Patent Owner argues that Petitioner provides no reason or evidence for how or why a skilled artisan would have been motivated to combine Culbert's task resource utilization vector with Boland's agent in the claimed manner. Prelim. Resp. 53. Patent Owner argues that Petitioner's conclusory assertions that one of ordinary skill could have adopted the Culbert task resource utilization vector in Boland, or that Culbert satisfied unspecified “implementation details” lacking in Boland, are an insufficient basis for the combination. *Id.* at 53–54. Patent Owner submits that Boland's workload manager 116 distributes resources based on resource usage by partitions, and has no missing details that need to be remedied by Culbert. *Id.* at 54. Moreover, Patent Owner points out that the task resource utilization vector in Culbert resides in the task or resource manager 170, which Petitioner equates to the '005 patent's applications and first manager process, and so there would be no reason to insert the task resource utilization vector into Boland's agents, which Patent Owner equates to the second manager process. *Id.* at 55–56 (citing Ex. 1005, 7:61–65).

c) Analysis

We agree with Patent Owner that Petitioner has not sufficiently shown that several claim 8 requirements are taught or suggested by the combination of Boland and Culbert. In particular, we are not persuaded that Boland and Culbert, taken alone or in combination, render obvious the required first and second manager processes, where the first manager process reallocates resources in response to requests from second manager processes in each computer domain. Culbert, which is a single multitasking system not made up of multiple domains, reallocates resources to tasks using a single manager process — resource manager 170. Ex. 1005, Fig. 1, 3:16–18, 5:31–36, 6:51–54. Likewise, Boland’s scheduler 280, reallocates resources based on rule information stored in the scheduler for allocating processes including a minimum resource allocation and the priorities for processing allocation for each process. Ex. 1004, Fig. 4, 6:65–7:18. Petitioner characterizes the Culbert resource manager as “centrally control[ing] resource allocation” (Pet. 22), and argues that this would motivate adopting scheduler 280 of Boland to such a central role, with Boland’s agents sending to it requests for resources. Pet. 55. But the resource manager of Culbert serves a multitasking system that is not made up of multiple domains, and therefore the resource manager has no relationship with other managers that could be characterized as “central,” and thus would not have served as a motivation to fundamentally change the approach of Boland.

Moreover, Petitioner’s reliance on the agents of Boland as serving the role of the required second manager process is misplaced. Pet. 56–58. The sole function of Boland’s agents is to supply a message to domain resource manager DM (106c), which permits the exclusive use of a computer

resource to client CLT by domain resource manager DM. Ex. 1004, 5:63–6:2. The agents have no role in requesting resources, and do not function “in response to analysis of . . . performance data,” given that the alleged source of such data, the local resource managers, transmit performance data to the domain resource manager, not the agents. *Id.* at 5:49–52, 6:3–6. Nor do we see any other reason to fundamentally change the operation of the Boland agents.

In addition, we are not persuaded that there is sufficient motivation to modify Boland’s agents to include Culbert’s resource utilization records, which Petitioner equates to the list of application priority levels and resources needed to meet the priority levels as per requirement 8.4 of claim 8. In Culbert, the resource utilization records are included in the task structure itself, the resource manager 170, or media operating system 180. Ex. 1005, 7:61–63. Therefore, if anything, Boland would have been modified to include the resource utilization records in the applications or scheduler 280, which are the alleged counterparts to the Culbert tasks and resource manager. In addition, we do not understand the basis for Petitioner’s argument that the run levels of the resource utilization records correspond to the required application priority levels, given that Culbert discloses an entirely separate “task list,” with entries including the priorities of each task, and which has nothing to do with run levels. Ex. 1005, Fig. 4, 9:57–10:2.

Accordingly, for the reasons set forth above, we determine that the Petition does not demonstrate a reasonable likelihood that Petitioner would prevail in establishing that claim 8 would have been obvious over the combination of Boland and Culbert.

G. Alleged Obviousness of Claims 8–14 Over Boland, Culbert, and Bugnion

Petitioner challenges claims 8–14 as obvious over the combination of Boland, Culbert, and Bugnion. Pet. 66–74. For claim 8, Petitioner’s challenge relies on Bugnion as modifying the combination of Boland and Culbert to implement hosts C–F of Boland as virtual machines. *Id.* at 67–68. Even assuming such a combination is justified, nevertheless our determination that the Petition does not demonstrate a reasonable likelihood that Petitioner would prevail in establishing that independent claim 8 would have been obvious over the combination of Boland and Culbert remains unchanged. Accordingly, we likewise determine that the Petition does not demonstrate a reasonable likelihood that Petitioner would prevail in establishing that dependent claims 8–14 would have been obvious over the combination of Boland, Culbert, and Bugnion.

IV. DISCRETIONARY DENIAL OF INSTITUTION

Patent Owner argues that we should exercise our discretion to deny institution because “the ’005 Patent is the subject of three petitions and nine duplicative grounds based on recycled references filed by two parties who have been coordinating efforts for years to needlessly increase costs to Patent Owners.” Prelim. Resp. 17. Patent Owner refers to this proceeding as well as IPR2022-01406 and IPR2022-01408, both brought by Google LLC. *Id.* Patent Owner argues that the IPRs are “needlessly complicated and duplicative,” and that under these circumstances the Board has discretion to deny institution of the petition. *Id.*

Patent Owner also argues that institution should be denied pursuant to 35 U.S.C. § 325(d) because the petition presents substantially the same art

(in particular, the Eilert reference) and arguments previously considered during prosecution of the '005 application. Prelim. Resp. 20–26.

However, because we are denying institution on the merits, it is unnecessary to determine whether it is appropriate to exercise our discretion to deny institution based on Patent Owner's arguments.

V. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that the Petition is denied and no *inter partes* review is instituted.

IPR2022-01399
Patent 7,748,005 B2

For PETITIONER:

Michael H. Jones
Mark T. Rawls
ROTHWELL, FIGG, ERNST & MANBECK, P.C.
mjones@rfem.com
mrawls@rfem.com

Roshan S. Mansinghani
Jordan Rossen
UNIFIED PATENTS, LLC
roshan@unifiedpatents.com
jordan@unifiedpatents.com

For PATENT OWNER:

Hong Annita Zhong
Amy E. Proctor
Blair Silver
IRELL & MANELLA LLP
hzhong@irell.com
aproctor@irell.com
bsilver@irell.com