IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

JUNIPER NETWORKS, INC.,

Petitioner,

v.

SWARM TECHNOLOGY, LLC,

Patent Owner.

Case IPR2021-01445

U.S. Patent No. 9,852,004

PETITIONER'S REPLY

TABLE OF CONTENTS

I. INTRODUCTION

Patent Owner's Response provides no evidence or arguments showing the Board erred in concluding that the Leong-AppleTalkBook combination teaches or renders obvious every limitation of the challenged claims.

PO concedes AppleTalkBook teaches dynamically accepting components into a system on a plug-and-play basis. But PO argues those teachings wouldn't have motivated a POSITA to have Leong accept co-processors on a plug-and-play basis without communication with the controller.[1] PO ignores the evidence that the Institution Decision identified as preliminarily establishing motivation to combine—i.e., avoiding the need for manual system configuration and allowing Leong's bulletin board to automatically connect to added processing units.[2]

PO also premises its patentability arguments on limitations imported from the specification. But PO agreed that no claim construction was necessary and that each term maintains its ordinary and customary meaning.[3] And it's well-settled law that reading limitations from a preferred embodiment into the claims is improper, absent a clear indication in the intrinsic record that PO intended to limit the scope

[1] POR, 31-32.

[2] Paper 15, Institution Decision ("Decision"), 19, 23-24.

[3] Pet., 11-12; POR, 6, 19-21.

of certain claim terms.[4] PO hasn't established any such indication in the intrinsic record—and indeed, none exists. Thus, nothing supports PO's efforts to import limitations through the claim terms "controller," "task," "without any communication with the controller," "retrieve," "deliver," "descriptor," and "notify."[5]

Because the prior art continues to disclose or render obvious each claim element, Petitioner respectfully asks the Board to find claims 1-12 unpatentable.

II. Leong Discloses Claim Elements 1.2 and 3.2's "Controller."

A proper reading of the claims shows the Board correctly found that Leong's surveying unit discloses the claimed "controller." Indeed, PO's arguments to the contrary require an improper importation of limitations and misinterpretation of Leong.

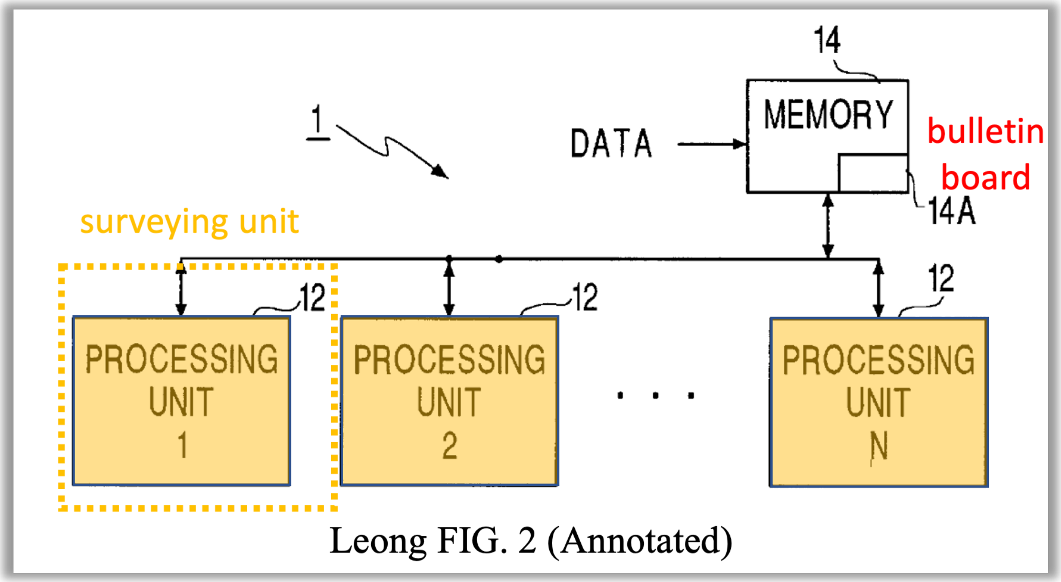
Claim elements 1.2 and 3.2 each recite "a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks."[6] The

[4] *EPOS Techs. Ltd. v. Pegasus Techs. Ltd.*, 766 F.3d 1338, 1341 (Fed. Cir. 2014) (quoting *Liebel-Flarsheim Co. v. Medrad, Inc.,* 358 F.3d 898, 913 (Fed. Cir. 2004)); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1327 (Fed. Cir. 2002). And a particular embodiment may not be read into a claim when the claim language is broader than the embodiment. *SuperGuide Corp. v. DirecTV Enters., Inc.*, 358 F.3d 870, 875 (Fed. Cir. 2004).

[5] POR, 18 (controller), 21 (task), 33-34 (without any communication with the controller), 26-27 (retrieve), 27 (deliver), 21-22 (descriptor), 54 (notify).

[6] '004 Patent, claims 1, 3.

Board found that Leong discloses that one processing unit 12 acts as a surveying unit to meet the claimed "controller."[7] And the Board determined that Leong's controller (i.e., surveying unit) populates the bulletin board with tasks.[8]



Leong FIG. 2 (Annotated)

Leong's Figure 2 shows bulletin board 14a is a region of memory 14.[9] Leong's bulletin board 14a and memory 14 store all tasks, status information, and electronic data that each co-processor executes when processing tasks.[10] Leong teaches the controller accepts new data into memory 14 when that data satisfies

[7] Decision, 26.

[8] *Id.*; Pet., 13-14 (citing Leong, 4:9-24, 6:16-39, 8:65-9:4, Figs. 5-6; Weissman, ¶47).

[9] Pet., 22 (citing Leong, 3:9-11; Weissman, ¶80).

[10] *Id.*, 22 (citing Leong, 3:9-11; Weissman, ¶80); Nelson Tr. (Ex. 1028), 149:1-4 (reading tasks from memory), 184:15-18 (reading data from memory).

certain review criteria.[11] Then Leong performs another review of each task's status information, including task type, priority, and conditions, when posting to and updating the bulletin board.[12]

Leong's controller (i.e., surveying unit) posts tasks.[13] When executing tasks, Leong's processing units read the bulletin board for posted tasks and the task's status information.[14] Leong states that specific tasks aren't critical "so long as the overall work to be performed on the electronic data may be partitioned into the tasks."[15] In one example, Leong groups tasks into accepting, sorting, and distributing types.[16] And Leong's tasks "may be further broken down into smaller tasks if desired."[17]

[11] Leong, 6:25-35.

[12] Pet., 26; Leong, 6:36-46.

[13] Leong, 1:55-67 ("an electronic bulletin board for posting the one or more tasks"), 3:9-18 ("The tasks… are posted on an electronic bulletin board 14a"); 6:16-18 (describing processing units posting tasks), 6:18-20 (describing a specialized agent posting tasks).

[14] Pet., 22, 28 (citing Leong, 6:47-61; Weissman, ¶89).

[15] *Id.*, 27 (citing Leong, 2:54-64).

[16] *Id.*

[17] *Id.*; *see also id.*, 22.

PO doesn't dispute these Leong teachings. And PO's expert Dr. Nelson agrees that Leong posts tasks.[18] Instead, PO asserts that the claimed "controller" requires dividing computational problems into groups of tasks which the controller then populates to the task pool.[19] But the claims don't include these additional requirements and PO hasn't proposed a construction for "controller" (or any other claim term).[20] As Dr. Weissman explains, Leong discloses a controller under that term's plain and ordinary meaning because Leong's surveying unit is a processor that populates the task pool with tasks.[21]

Aside from attorney argument that the claimed "controller" is "qualitatively different" from Leong, PO doesn't explain how Leong's disclosures differ from PO's imported requirements for the "controller."[22] As Dr. Weissman states, Leong

[18] Nelson (Ex. 2007), ¶152 ("one of the micro-processing units 12 executes the posted task").

[19] POR, 19-20 (citing 6:40-48, not claim 1).

[20] Weissman Tr. (Ex. 2010), 244:11-15 ("I don't believe there's anything in plain and ordinary meaning or customary meaning that would require [dividing computing requirements into threads]").

[21] Pet., 24-28; Weissman, ¶¶84-91.

[22] *See* Leong, 2:54-64.

"is decomposing or partitioning, any words you want to use, of the problem into tasks" and populating those tasks into the task pool.[23]

PO also argues Leong's controller (i.e., surveying unit) only looks up customer profiles and doesn't post tasks.[24] PO ignores that the Leong sentence it cites further says the controller "determine[s] what tasks must be posted and formulate[s] the status information for the tasks."[25] Here, Leong discloses posting tasks regarding check images, statements, and product marketing material, based on customer profile information.[26] Leong's controller assigns higher priorities to certain tasks based on customer information, such as the processing of exception checks.[27] While in one embodiment, Leong teaches its controller looks up customer information, Leong's controller isn't limited to that embodiment.[28]

Similarly, PO argues Leong's controller (i.e., surveying unit) only posts status information to the task pool and asserts Leong's Figure 3 confirms "TASK_N" is

[23] Weissman Tr. (Ex. 2010), 218:17-23.

[24] POR, 20-21 (citing Leong, 9:1-4); *but see* Nelson (Ex. 2007), ¶152 ("one of the micro-processing units 12 executes the posted task").

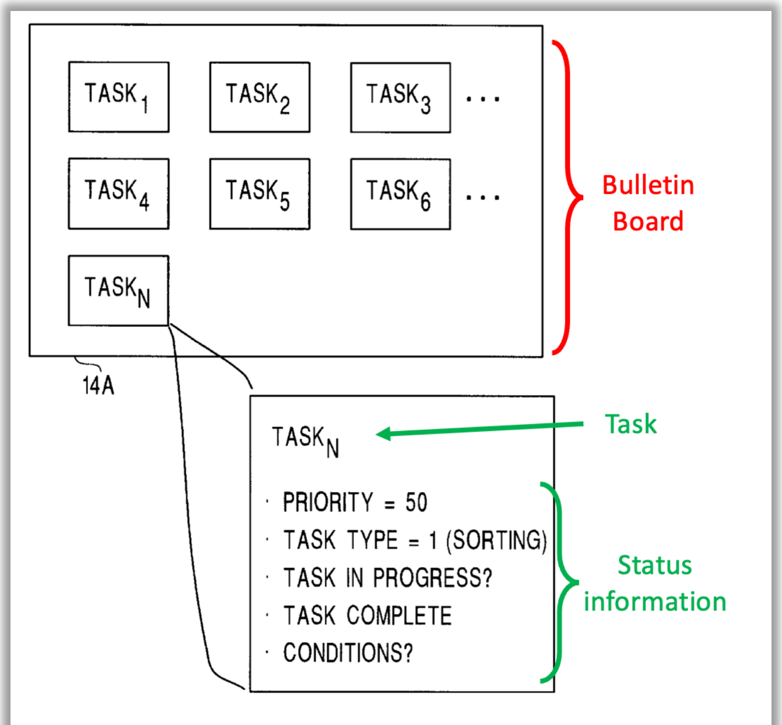
[25] Leong, 9:1-4.

[26] *Id.*, 10:18-30.

[27] *Id.*, 9:32-54 (describing priority levels of tasks).

[28] *In re Lamberti*, 545 F.2d 747, 750 (C.C.P.A. 1976) ("[A]ll disclosures of the prior art, including unpreferred embodiments, must be considered").

shorthand for task status information.[29] But this ignores multiple disclosures that Leong's controller also posts the tasks themselves to memory and that the tasks form part of the contents of the bulletin board.[30] And Leong's Figure 3 separately depicts both TASK$_N$ and the status information (e.g., "task in progress?") demonstrating TASK$_N$ means something more than status information.[31]



Leong FIG. 3 (Annotated)

[29] POR, 21-25.

[30] *E.g.*, Leong, 3:6-19 (discussing tasks "posted on an electronic bulletin board"), 6:47-61 (discussing reading "bulletin board 14a for posted tasks and the status information"), 7:1-13 (same), 7:35-61 (describing posted tasks), 8:65-9:4 (same).

[31] *Id.*, Fig. 3 (exploded view of TASK$_N$).

Indeed, Leong confirms tasks, not status information, are designated through TASK_N and that these tasks are part of the bulletin board's contents.[32] And Leong, a U.S. patent, includes claims reciting posting tasks separately from status information.[33] Given these separate recitations, a POSITA would've understood the tasks posted to Leong's bulletin board are distinct from the posted task information.[34]

Even under PO's narrow interpretation of Leong, PO and Dr. Nelson acknowledge Leong stores data in memory 14 for execution by the co-processors when performing tasks.[35] And PO and Dr. Nelson acknowledge bulletin board 14a is part of memory 14.[36] To the extent PO attempts to distinguish between bulletin board 14a and memory 14, PO doesn't contest Leong's bulletin board meets the claimed "task pool" and that bulletin board 14a is a portion of memory 14 overlayed with the task pool and status information.[37] Thus, even under PO's view of

[32] *Id.*, 3:6-19 ("The tasks, generally designated TASK_1 through TASK_N, are posted on an electronic bulletin board").

[33] *Id.*, claims 1 ("posted tasks"), 2 ("status information").

[34] *Ethicon Endo-Surgery, Inc. v. U.S. Surgical Corp.*, 93 F.3d 1572, 1579 (Fed. Cir. 1996) (noting claim terms are presumed to have different meanings).

[35] Nelson Tr. (Ex. 1028), 150:5-9, 169:12-18, 184:15-18.

[36] *Id.*, 140:19-22.

[37] Leong, 3:9-11; Pet., 22-23.

Leong, Dr. Nelson confirmed a POSITA would've understood bulletin board 14a associates task status information with "corresponding" task data in memory 14.[38]

Thus, Petitioner establishes that Leong's controller is configured to populate the task pool with multiple tasks.[39]

III. LEONG'S "READING" DISCLOSURES DESCRIBE CO-PROCESSORS THAT MEET CLAIM ELEMENTS 1.3-1.4 AND 3.3-3.4.

Claim elements 1.3-1.4 and 3.3-3.4 require the co-processors "retrieve" tasks from the task pool and "deliver" them to the respective co-processor. PO argues each task is pre-loaded into each of Leong's co-processors because Leong refers to "reading" the bulletin board.[40] PO posits that a POSITA would've only understood "reading" to mean reading all the bulletin board's contents into each co-processor's cache memory, and thereafter processing the status information to identify a task for execution.[41]

PO's argument, however, contradicts Leong's teachings that its co-processors read the bulletin board's tasks.[42] Leong's co-processors also determine

[38] Nelson Tr. (Ex. 1028), 184:15-18.

[39] Pet., 24-28.

[40] POR, 27-28.

[41] *Id.*

[42] Pet., 30; Leong, 3:11-13.

"whether they are capable of performing one or more tasks."[43] Leong repeatedly states that "when they read the bulletin board," the co-processors determine what tasks to execute by searching the task pool.[44] These determinations occur "when" Leong's read occurs, meaning Leong's read is more than just reading data from memory 14 to the co-processor's cache memory as PO suggests.[45] This is because a POSITA would've understood processing capability exists in memory 14 for performing Leong's determinations.[46]

As Leong shows in Figure 2, an Ethernet network connects co-processors 12 and memory 14.[47] In such a system, Dr. Weissman confirms a POSITA would've understood Leong's memory has processing features for accessing the network in similar fashion as the controller and co-processors.[48] And PO's expert

[43] Pet., 30; Leong at 3:13-15; Weissman Suppl., ¶4.

[44] Pet., 30; Leong at 3:27-31, 3:32-35, 3:52-54, 3:59-61; Weissman Suppl., ¶¶4-5.

[45] Weissman Suppl., ¶¶5-6.

[46] Weissman Tr. (Ex. 2010), 176:12-18 ("It has the processing capacity to receive a request and to do searching"); Weissman Suppl., ¶7.

[47] Pet., 39 (citing Leong, 5:21-25); Weissman Suppl., ¶7.

[48] Weissman, ¶¶111-112; *see also* Weissman Tr. (Ex. 2010), 24:1-25:24, 25:7-8 ("I think the bulletin board is accessible on the network"); Weissman Suppl., ¶7.

acknowledges that memory connected to a network includes processing features since "[m]emories do not hang off of networks on their own."[49]

Contradicting these teachings, PO asserts that each of Leong's co-processors operates against a local copy of the bulletin board.[50] But under PO's theory, the system would need to transmit copies to each processing unit, increasing network traffic.[51] And conflicts would occur frequently due to the nature of the data in the bulletin board (e.g., tasks and status information) where multiple co-processors make independent determinations and execute tasks in parallel against their independent copies of the entire bulletin board.[52] For example, while one co-processor determines which task to execute, another co-processor changes one of the bulletin board's tasks which would've affected the first co-processor's determination, thereby leading to system inconsistencies.[53] PO's proposed meaning of Leong's "reading" isn't consistent with how a POSITA would've implemented Leong because it compounds the problems associated with a distributed computing system.[54]

[49] Nelson Tr. (Ex. 1028), 205:19-20.

[50] POR, 27-28, 46.

[51] Weissman Suppl., ¶¶8-10

[52] *Id.*; *see* Leong, 1:55-67.

[53] Weissman Suppl., ¶¶9-10.

[54] *Id.*

This is contrary to Leong's goal of creating a "failure proof" system.[55] And if PO's theory were correct, a POSITA would expect Leong to explain how and when to reconcile the potential conflicts among the processing units after task completion.[56] But Leong doesn't include such disclosures, further showing PO's theory is incorrect.

As Dr. Weissman explains, a POSITA would've understood Leong's designs avoid such problems.[57] For example, Leong's task pool provides each co-processor only with a portion of the overall work (i.e., a task), and each co-processor executes their task.[58] Leong's system doesn't reconcile multiple parallel versions and changes because Leong's task pool prevents multiple co-processors from executing the same task at the same time.[59] In doing so, a POSITA would've understood that Leong's co-processor don't operate against a local copy of the bulletin board, as PO and Dr. Nelson suggest.[60] Rather, Leong's co-processors retrieve tasks from the bulletin board (i.e., task pool), deliver the task to the co-processor, process that

[55] Leong, 5:25-35.

[56] Weissman Suppl., ¶¶9-10.

[57] *Id.*, ¶11.

[58] *Id.*

[59] *Id.*

[60] *Id.*, ¶¶11-12.

task to generate resulting data, and update the bulletin board upon task completion.[61]

PO argues "[t]he '004 Patent's division and classification of tasks… enables corresponding first and second co-processors to retrieve and deliver their respective tasks proactively," and that Leong lacks this.[62] PO improperly imports additional requirements absent from the claims. As Section II above and the Petition detail, Leong offers the same division and classification of tasks as the '004 Patent.[63]

Accordingly, Leong's processing units disclose or render obvious the co-processors in claim elements 1.3-1.4 and 3.3-3.4.

IV. THE LEONG-APPLETALKBOOK COMBINATION RENDERS OBVIOUS CLAIM ELEMENT 1.5'S DYNAMIC ACCEPTANCE OF CO-PROCESSORS.

The Leong-AppleTalkBook combination renders obvious claim element 1.5. Petitioner demonstrates that a POSITA would've been motivated and found it obvious to modify Leong with AppleTalkBook's teachings so that additional processing units join the system without the user needing to manually configure each co-processor.[64] PO argues a POSITA wouldn't have been motivated to incorporate

[61] *Id.*

[62] POR, 26-27.

[63] Pet., 24-28; *see* Weissman Suppl., ¶8.

[64] Pet., 35-41.

Apple's EtherTalk cards and protocols into Leong's system because Leong doesn't contemplate adding new co-processors.[65] But this misreads Leong, misinterprets Petitioner's obviousness argument, and fails to consider what the combined teachings of Leong and AppleTalkBook would've disclosed to a POSITA.[66]

A. Leong Confirms Adding Co-Processors was Known.

PO argues Leong only discloses reducing the number of co-processors.[67] But Leong's scalable system seeks to address prior art approaches to adding new co-processors.

Leong states that it was known in the art to have "readily scalable" systems able to accommodate "configuration changes when other processing units 12' are added to the system 1'."[68] PO's expert admits the notion of scalability, which Leong focuses on improving, involves adding processors.[69] And PO's expert admits Leong's scalability discussion confirms a POSITA would've known how to

[65] POR, 38-39.

[66] *See In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012); *In re Keller*, 642 F.2d 413, 425 (Fed. Cir. 1981); *In re Nievelt*, 482 F.2d 965, 968 (C.C.P.A. 1973); *In re Sneed*, 710 F.2d 1544, 1550 (Fed. Cir. 1983).

[67] POR, 30.

[68] Leong, 1:44-52, Fig. 1 (disclosing prior art system 1' with processing unit 12').

[69] Nelson Tr. (Ex. 1028), 89:16-92:13.

add processors to systems.[70] Further, Leong teaches co-processors failing while the systems maintains throughput, which could involve adding a co-processor.[71]

Moreover, Leong discloses idling co-processors if the number of posted tasks of a certain type drops beneath a threshold.[72] Leong teaches that, after a predetermined period, "the unit would again be available to perform posted tasks of a certain type," thereby restoring the idled processing power.[73] Leong also discloses co-processors switching the types of tasks executed, thereby adding co-processors for certain task types at the expense of others.[74] These all show Leong isn't limited to only removing co-processors, as PO alleges.

Thus, Leong doesn't teach away from adding co-processors into the processing system, as required by claim element 1.5.

B. Petitioner Doesn't Propose Physically Incorporating AppleTalkBook Into Leong's System.

PO argues Leong teaches away from AppleTalkBook because Leong states that hardware with custom designs and special network switching technology has

[70] *Id.*, 160:8-22.

[71] Leong, 5:29-34; Nelson Tr. (Ex. 1028), 192:12-193:7 (agreeing "may be affected" includes situations which aren't affected).

[72] Pet., 36 (citing Leong, 4:50-5:11; Weissman, ¶105).

[73] *Id.*

[74] *Id.*

"disadvantages."[75] This is incorrect. Leong's discussion of custom hardware, such as "IBM SP2, NCR, and Cray systems" as having "disadvantages" doesn't teach away from using plug-and-play features like those taught in AppleTalkBook.[76] Apple's systems are mass-market personal computers, nothing like custom-designed SP2, NCR, and Cray systems that Leong references.[77] And PO hasn't shown how Leong criticizes, discredits, or otherwise discourages accepting co-processors on a plug-and-play basis due to AppleTalkBook's teachings.[78]

PO alleges there's no reasonable expectation of success because of "known, and unknown, consequences" from the Leong-AppleTalkBook combination.[79] But PO only identifies consequences arising from its faulty assumption that Petitioner proposes physically incorporating Apple's hardware/software into Leong's system. Petitioner's combination doesn't propose this; rather, Petitioner establishes that a

[75] POR, 38-39.

[76] Leong, 1:25-27; Weissman Suppl., ¶13.

[77] Weissman Suppl., ¶13.

[78] *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 567 F.3d 1314, 1327 (Fed. Cir. 2009) (quoting *In re Fulton*, 391 F.3d 1195, 1201 (Fed. Cir. 2004)) ("A reference does not teach away... if it… does not 'criticize, discredit, or otherwise discourage' investigation into the invention claimed.")

[79] POR, 41.

POSITA would've been motivated to apply AppleTalkBook's teachings.[80] For example, Petitioner shows that AppleTalkBook describes network-connected devices that transmit broadcast packets to implement plug-and-play functionality.[81] PO and Dr. Nelson don't contest that by 2013 dynamically accepting devices into a system on a plug-and-play basis was a well-known concept. And the Petition establishes that a POSITA comprehensively understood how to implement plug-and-play—as illustrated by AppleTalkBook's teachings—into systems interconnected over an Ethernet network, like Leong's.[82] This demonstrates that a POSITA would've had a reasonable expectation of success in combining the teachings of Leong and AppleTalkBook.[83]

Further, Petitioner's EtherTalk analysis demonstrates that a POSITA would've had a reasonable expectation of success in modifying Leong's Ethernet-connected system to accept a co-processor on a plug-and-play basis.[84] But the

[80] Pet., 35-41 (citing Weissman, ¶¶105-115).

[81] *Id.*, 39-41 (citing Weissman, ¶¶111-115).

[82] *Id.*, 38 (citing Ex. 1003, ¶110) (explaining the combination removes the "need for manual configuration to add new processing units" and "wouldn't require any [central] coordination"), 35-41.

[83] *Intelligent Bio-Sys., Inc. v. Illumina Cambridge, Ltd.*, 821 F.3d 1359, 1367 (Fed. Cir. 2016).

[84] Pet., 39-40.

Leong-AppleTalkBook combination doesn't require EtherTalk or other protocols that AppleTalkBook describes.[85] Indeed, contrary to PO's assertions, nothing in Petitioner's combination requires the physical incorporation of the AppleTalk protocol stack[86] or the AppleTalk packets into Leong's system.[87] Rather, Petitioner demonstrates that a POSITA would've understood how to use any well-known plug-and-play protocol in Leong's system, and would have a reasonable expectation of success in doing so.[88] AppleTalkBook confirms that more than 20 years prior to the '004 Patent, a POSITA would've known of plug-and-play techniques.[89] Applying AppleTalkBook's teachings to Leong to implement plug-and-play wouldn't require significant reconfiguration of Leong, and was within the routine skill of a POSITA.[90]

At bottom, PO's arguments and Dr. Nelson's opinions focus strictly on physically incorporating EtherTalk/AppleTalk hardware and software into Leong's system. They don't address whether the collective teachings of Leong and AppleTalkBook would've motivated a POSITA to dynamically accept a co-processor

[85] *Id*., 35-41.

[86] POR, 41; Weissman Suppl., ¶14.

[87] POR, 42-43; Weissman Suppl., ¶14.

[88] Weissman Suppl., ¶¶14-16.

[89] *Id*.

[90] *Id*.

into the system on a plug-and-play basis.[91] Thus, Petitioner establishes that the Leong-AppleTalkBook combination renders obvious claim element 1.5.

C. PO's Broadcast Packet Argument is Incorrect.

PO agrees that "the dynamic configuration of the AppleTalk protocol can occur 'without a system administrator'" and that AppleTalk sends identification information via broadcast packets.[92] But PO argues that any system having a node transmitting broadcast packets (or any other transmission) to all other nodes doesn't meet the "without any communication" requirement of claim element 1.5.[93]

PO's arguments again improperly physically incorporate Apple hardware/software into Leong's system and further posit that Petitioner's obviousness combination requires each co-processor send identification information to the controller. Yet PO agrees that only "[i]n some instances" would Leong's controller respond to the broadcast packet, i.e., when the newly added device chooses a used node number.[94] So even under PO's theory, sometimes the controller never responds to a new co-processor's broadcast packet.[95] Because networks commonly use broadcast packets to distribute control information, such as routing

[91] *Id.*

[92] POR, 32.

[93] *Id.*, 30-35; Nelson (Ex. 2007), ¶129 (including broadcasts as communication).

[94] POR, 33; Nelson (Ex. 2007), ¶129.

[95] AppleTalkBook, 108-109.

information, PO won't expressly construe "without any communication" as precluding a system having nodes sending broadcast packets that receive no response.[96] But a POSITA wouldn't equate broadcasts and communications because broadcasts aren't two way.[97] And downstream devices may, for example, filter and drop broadcast packets to reduce CPU load.[98] The '004 Patent's plug-and-play and the communication channel disclosures show a co-processor (cell 12) sending packets to the controller's task pool and the controller responding with an acknowledgment message.[99] This tracks with a POSITA's understanding of the distinction between broadcasting messages versus network devices communicating with one another.[100]

The Petition gives one exemplary implementation where Leong's modified system provides plug-and-play functionality by broadcasting identification

[96] *Id.*, 108.

[97] *Id.*, 45 ("[C]ommunications between [networks]"), 83 ("[M]any users can share resources and communicate with each other"), 42 ("[C]ommunication between any two devices on either network"), 91 ("[C]ommunications take place between adjacent layers"), 93 ("[C]ommunications between devices"); Weissman Suppl., ¶¶17-19.

[98] Weissman Suppl., ¶18.

[99] '004 Patent, 4:58-5:16, 6:18-33; Weissman Suppl., ¶19.

[100] Weissman Suppl., ¶19.

information, consistent with this understanding of broadcasting versus communicating. And PO doesn't address how AppleTalkBook avoids scenarios where "broadcast packets and their responses could flood the network" if they traversed every connection to every node.[101] AppleTalkBook teaches establishing zones dividing systems into logical groups.[102] PO doesn't address that AppleTalkBook describes the benefits of having "groups that work closely together are in the same zone" and recommends zones "because of the network traffic optimization they provide."[103]

When electing to implement AppleTalkBook's plug-and-play teachings in Leong's system by utilizing AppleTalk's broadcast packet, AppleTalkBook confirms a POSITA would've been motivated to avoid any communication with Leong's controller and co-processors by applying AppleTalkBook's zone teachings to group devices that "work closely together."[104] This would result in Leong's system having a zone for the controller(s) separate from the dynamically accepted co-processor(s), thereby limiting broadcast traffic emitted from the dynamically

[101] AppleTalkBook, 221; Weissman Suppl., ¶20.

[102] AppleTalkBook, 220; Weissman Suppl., ¶20.

[103] AppleTalkBook, 220; Weissman Suppl., ¶20.

[104] AppleTalkBook, 220; Weissman Suppl., ¶20.

added co-processors.[105] The dynamically added co-processors share a zone with the memory with the task pool so that processing capability is dynamically added without broadcasting packets to or communicating with the controller.[106] Here, because of the zones, the broadcast packets emitted by the newly added co-processors only reach other newly added co-processors and the memory with the task pool.[107] This suffices for adding new co-processors into the system because the added Leong co-processors need only identify the task pool to obtain work.[108] A POSITA would've been motivated to make this modification because it allows for dynamic acceptance of co-processors that need only contact the task pool to obtain work, minimizing the configuration needed to use additional processing capacity, and preventing broadcast flooding.[109]

V. THE LEONG-APPLETALKBOOK COMBINATION RENDERS OBVIOUS CLAIMS 2 AND 3'S "AGENTS."

The '004 Patent discloses that "an agent is generally analogous to a data frame in the networking sense, in that an agent may be equipped with a source

[105] AppleTalkBook, 220; Weissman Suppl., ¶20.

[106] Weissman Suppl., ¶20.

[107] AppleTalkBook, 220; Weissman Suppl., ¶20.

[108] Weissman Suppl., ¶20.

[109] *Id*.

address, a destination address, and a payload."[110] PO incorrectly alleges Petitioner doesn't "map any feature of Leong to the claimed agent" recited in claim 2 and 3.[111] But Petitioner details how Leong discloses or renders obvious using Ethernet data frames that are agents.[112] PO doesn't contest this but argues that the claimed "agents" require something more.[113] But nothing in the intrinsic record requires additional requirements for an "agent" beyond that term's plain and ordinary meaning.

For claim 2's "agent" searching the task pool for tasks of certain types, Leong's status information includes task type (i.e., indicia of a first task type), and Leong's co-processors "search the bulletin board (i.e., task pool) for task-types matching their abilities."[114] And as discussed above in Section III, Leong's reading involves determinations that are more than the mere reading from memory that PO argues. Thus, Petitioner demonstrates that Leong discloses or renders obvious claim 2's "agent."[115]

[110] '004 Patent, 8:30-33.

[111] POR, 46-48.

[112] Pet., 41-44 (claim 2's agents), 46-53 (claim 3's agents).

[113] POR, 45-48 (claim 2's agents), 48-53 (claim 3's agents' characteristics).

[114] Pet., 41-44.

[115] '004 Patent, claim 2.

Like claim 2, claim 3's "agent" searches the task pool. It also retrieves a task and then returns from the task pool. PO concedes Leong's Ethernet-connected system includes data frames with the same fields as claimed.[116] PO, however, argues that because "the Ethernet Standard is agnostic as to the contents of the Data field," what the fields communicate distinguish the claims.[117] Thus, PO's distinction relies on the content of the agent's data fields.[118]

But the printed matter doctrine specifies no patentable weight applies to printed matter in claim elements unless the printed matter functionally relates to the substrate onto which the printed matter is applied.[119] Here, the field's contents don't have a functional or structural relationship to anything aside from what the Ethernet Standard already prescribes for the use of values in the fields.[120] Nothing

[116] POR, 48 ("While the data frame of the Ethernet Standard does include fields corresponding to a Source, Destination, and Data").

[117] POR, 48-53 (arguing the claimed contents of the first payload).

[118] *In re Distefano*, 808 F.3d 845, 848-49 (Fed. Cir. 2015) ("The first step of the printed matter analysis is the determination that the limitation in question is in fact directed toward printed matter").

[119] *Id.*; *Praxair Distribution, Inc. v. Mallinckrodt Hosp. Prods.*, 890 F.3d 1024, 1031 (Fed. Cir. 2018).

[120] *Distefano*, 808 F.3d at 850; *Praxair*, 890 F.3d at 1031; *Ex parte Nehls*, 2008 WL 258370, at *7-8 (B.P.A.I. 2008) (precedential) (finding claimed descriptive material didn't affect performance of prior-art method).

claimed or described in the specification suggests the processing of the source or destination addresses of the claimed "agent" differs from those fields of the Ethernet data frame.[121] And PO doesn't dispute that other information, including the functions the co-processors perform, would exist in the data frame's payload field in Leong's system.[122]

These aren't hindsight modifications to Leong's system, but are how a POSITA would've used these fields, as Dr. Weissman explains.[123] Leong's Ethernet data frame includes a payload field that allows "any arbitrary sequence of octet values."[124] PO hasn't shown the claimed system's payload contents provide a different result from what a POSITA would expect if Leong's data frame payload stored the function which the respective co-processor performs via the claimed "agent."[125]

Accordingly, the Leong-AppleTalkBook combination renders obvious the limitations relating to the claimed "agents."

[121] Pet., 52 (citing Weissman, ¶142).

[122] POR, 48.

[123] *Id.*; Weissman, ¶¶135-138.

[124] Weissman, ¶137.

[125] *Nehls*, 2008 WL 258370 at *8 (citing *Ex parte Curry*, 84 U.S.P.Q.2d 1272 (B.P.A.I. 2005)).

VI. THE LEONG-APPLETALKBOOK COMBINATION RENDERS OBVIOUS DEPENDENT CLAIM 9.

PO argues Leong doesn't disclose claim 9's task pool because Leong's bulletin board doesn't respond to the change in status information of TASK_{N} or affirmatively send a notification.[126] Claim 9 requires neither. Petitioner establishes that a POSITA would've understood Leong's changing of the task status information to "complete" is a notification, and that it would've been obvious to implement task completion notifications so that Leong's sequenced tasks execute appropriately.[127] Thus, Petitioner demonstrates the Leong-AppleTalkBook combination renders obvious claim 9.

Further, as Ground 3 details, Bates teaches it was known to notify controllers upon task completion. To the extent Leong lacks details regarding how different tasks signal completion before the execution of other tasks, Petitioner establishes that motivation existed to apply Bates' teachings regarding task synchronization in Leong's system.[128] Bates, for example, teaches notifying through polling of an identifier for completion, or through an interrupt sent to the recipient upon

[126] POR, 53-54.

[127] Pet., 67-73 (citing Weissman, ¶178).

[128] *Id.*, 83-86 (citing Weissman, ¶209).

completion.[129] Thus, the Leong-AppleTalkBook-Bates combination renders obvious claim 9 even with PO's unsupported sending requirement.

VII. PO's Criticisms of Dr. Weissman's Opinions Lack Merit.

PO asserts that hindsight drives Dr. Weissman's opinions,[130] that he didn't provide a response to those hindsight assertions,[131] and that he isn't clear on his use of the law of obviousness.[132] Each of these assertions lacks merit. Dr. Weissman opined that claims 1-12 are obvious over Leong and AppleTalkBook, claims 3-12 are obvious over Leong, AppleTalkBook, and the Ethernet Standard, and claim 9 is also obvious in further view of Bates's teachings.[133]

Dr. Weissman's declaration shows he performed an element-by-element analysis to determine whether each claim as a whole was obvious. PO cherry-picks a sentence from Dr. Weissman's declaration to assert he used an erroneous legal framework for his obviousness analysis.[134] But Dr. Weissman's 145 paragraphs of obviousness analysis bely PO's assertion and show he based his obviousness

[129] *Id.*, 84-85.

[130] POR, 15-16.

[131] *Id.*, 16.

[132] *Id.*, 17-18.

[133] Weissman Tr. (Ex. 2010), 244:21-245:5.

[134] POR, 17.

analysis on the proper legal framework.[135] The prior art provides the basis for his opinions, i.e., the usage of the Ethernet data frame fields in a conventional manner,[136] not hindsight. And while Dr. Weissman testified he drafted the entirety of his declaration,[137] Dr. Nelson testified he didn't write portions of his declaration and didn't review materials cited in his declaration. For example, Dr. Nelson admitted he didn't read or have knowledge about three legal decisions cited in his declaration.[138]

VIII. APPLETALKBOOK (EX. 1006) IS PRIOR ART.

The Board's rules require objections to the admissibility of evidence submitted pre-institution within ten business days of trial institution.[139] PO didn't timely object to the AppleTalkBook reference (Ex. 1006), and thus waived its admissibility argument.[140] And in its Response, PO incorrectly contends that AppleTalkBook wasn't publicly available.

[135] Weissman, ¶¶70-215.

[136] *Id.*, ¶141.

[137] *E.g.*, Weissman Tr. (Ex. 2010), 159:16-19 ("[T]he entirety of my declaration … was done at my direction. So it's my opinion").

[138] Nelson Tr. (Ex. 1028), 208:21-213:8.

[139] 37 C.F.R. §42.64(b)(1).

[140] *Qualcomm Inc. v. UNM Rainforest Innovations*, IPR2021-00377, Paper 66 at 9 (P.T.A.B. Jul. 15, 2022) (denying motion to exclude for untimely objections).

A. Library Records Confirm AppleTalkBook's Public Accessibility.

Dr. Sylvia Hall-Ellis, experienced in the field of library and information resources, confirmed the public accessibility of the "Hands-On AppleTalk" book using records created when a library received a copy of the work.[141] Vintage Apple, a digital repository for Apple-related documents, scanned the physical book and made a PDF, which Dr. Hall-Ellis also retrieved.[142] The PDF includes indicia matching those found in the Library of Congress's records, including the ISBN.[143] At her deposition, Dr. Hall-Ellis explained that she analyzed those records and the PDF to confirm the public accessibility and authenticity of AppleTalkBook.[144] For example, Dr. Hall-Ellis notes the PDF's completeness and seamless flow from one page to the next with no visible alterations.[145] And Vintage Apple's providing a PDF of the book is typical and doesn't affect its authenticity.[146]

[141] Ex. 1018, ¶¶47-53.

[142] *Id.*, ¶¶47-48; Ex. 2011, 70:20-71:1.

[143] Ex. 1018, ¶¶48-51; Ex. 2011, 64:24-66:7.

[144] Ex. 1018, ¶43; Ex. 2011, 28:16-29:11 (discussing authentic documents), 71:5-13 (confirming opinions on public accessibility).

[145] Ex. 1018, ¶48; *see* Ex. 2011, 54:1-18 (discussing duplicated cover), 56:7-15 (same), 57:12-58:9 (noting the PDF omitted blank pages).

[146] Ex. 1018, ¶48; Ex. 2011, 35:14-21 (identifying Vintage Apple's steps ensuring the availability of the best possible PDFs).

PO relies on the PDF's creation and appearance on the Internet Archive to challenge AppleTalkBook's prior art status.[147] But Dr. Hall-Ellis's opinion is to the public availability of "Hands-On AppleTalk," which is confirmed by the creation of library records in 1989.[148] These records contain matching indicia including the ISBN, copyright year, and publisher.[149] PO doesn't challenge any of this. Thus, Petitioner establishes that "Hands-On AppleTalk" is a printed publication publicly accessible before the '004 Patent's earliest possible priority date.

B. Ex. 1006 is Authentic.

PO also questions AppleTalkBook's authenticity, arguing that there was "no attempt to validate or even identify the creator" of the PDF (Ex. 1006).[150] But authenticity doesn't require an identification of the creator. Rather, Petitioner must only provide "evidence sufficient to support a finding that the item is what the proponent claims."[151] As the Board and numerous courts note, this burden is low.[152]

[147] POR, 9-10.

[148] Ex. 1018, ¶¶27-43; Ex. 2011, 16:1-13.

[149] Ex. 1018, ¶¶47, 49-51 (discussing library records with ISBNs).

[150] POR, 12-14.

[151] F.R.E. §901(a).

[152] *Fox Factory v. SRAM, LLC*, IPR2016-01876, Paper 59 at 63 (P.T.A.B. Apr. 2, 2018) (citing *United States v. Patterson*, 277 F.3d 709, 713 (4th Cir. 2002)); *United States v. Ceballos*, 789 F.3d 607, 617-18 (5th Cir. 2015) (noting "low" burden for authentication"); *United States v. Isiwele*, 635 F.3d 196, 200 (5th Cir.

PO hasn't raised any doubts as to the content of Ex. 1006.[153] And Dr. Hall-Ellis reviewed Ex. 1006 and confirmed its authenticity, noting that the text flows seamlessly and concluded the missing pages are blank.[154]

AppleTalkBook is therefore prior art to the '004 Patent.

IX. CONCLUSION

Ground 1 establishes that AppleTalkBook's teachings render obvious dynamically accepting a co-processor into Leong's system on a plug-and-play basis without any communication with a controller. And Grounds 1-3 show Leong teaches all remaining claim elements toward the claimed "controller," "co-processors," and "agents." Thus, Petitioner respectfully requests that the challenged claims be found unpatentable.

2011) (noting flaws in authentication go to weight not admissibility); *United States v. Gagliardi*, 506 F.3d 140, 151 (2d Cir. 2007) ("The burden of proof for authentication is slight. All that is required is a foundation from which the factfinder could legitimately infer that the evidence is what the proponent claims it to be.") (internal quotation marks omitted).

[153] Ex. 2011, 71:5-13 (confirming opinions on public accessibility).

[154] *Id.*, 57:12-58:9 (confirming omission of some blank pages but clarifying that no pages of "text, diagrams, charts, graphs, tables, index, appendices" are missing), 54:1-18 (noting Ex. 1006 includes duplicate book covers but that doesn't suggest incompleteness since covers don't relate to the book's content or alter the seamless flow of text).

Respectfully submitted,

/Joseph F. Edell/

Joseph F. Edell
Counsel for Petitioner

EXHIBIT LIST

Exhibit	Description
1001	U.S. Patent No. 9,852,004 ("'004 Patent")
1002	Prosecution History for U.S. Patent No. 9,852,004 ("'040 History")
1003	Declaration of Dr. Jon B. Weissman
1004	Curriculum vitae of Dr. Weissman
1005	U.S. Patent No. 6,006,249 ("Leong")
1006	M. Rogers & V. Bare, "Hands-On AppleTalk" (1989) (available at https://vintageapple.org/macbooks/pdf/Hands-On_Apple-Talk_1989.pdf, last accessed June 28, 2021) ("AppleTalkBook")
1007	"The Ethernet – A Local Area Network, Data Link Layer and Physical Layer Specifications," Version 1.0 (September 30, 1980) (available at https://gordonbell.azurewebsites.net/Digital/Ethernet%20Blue%20Book.pdf) (last accessed April 13, 2021) ("Ethernet," "the Ethernet standard")
1008	U.S. Application Publication No. 2007/0074207 ("Bates")
1009	"Comparison of Microsoft Windows versions" (available at https://en.wikipedia.org/wiki/Comparison_of_Microsoft_Windows_versions, last accessed March 18, 2021)
1010	"Official List Announcement: RC5-56 completion" (available at https://blogs.distributed.net/1997/10/22/00/00/bovine/, last accessed March 18, 2021)
1011	D. Anderson, *et al*. "High-Performance Task Distribution for Volunteer Computing," First IEEE International Conference on e-Science and Grid Technologies, pp. 5-8 (December 2005) (available at http://boinc.berkeley.edu/boinc_papers/server_perf/server_perf.pdf, last accessed March 18, 2021)
1012	Y. Shoham, "Agent-oriented programming," Artificial Intelligence, vol. 60, pp. 51-92 (1993) ("Shoham")
1013	M. Burgin, G. Dodig-Crnkovic, "A Systematic Approach to Artificial Agents," In Computer Science (2009) (available at http://arxiv.org/abs/0902.3513) ("Burgin")

Exhibit	Description
1014	U.S. Patent No. 9,189,281 ("Chen")
1015	James F. Kurose & Keith W. Ross, Computer Networking, A Top-Down Approach, pp. 469-473 (4th ed. 2008) ("Kurose")
1016	A. Oppenheimer, "A History of Macintosh Networking" (January 2004) (available at https://web.archive.org/web/20061016132614/http://www.opendoor.com/nethistory/MacWorld2004/index.html, last accessed June 30, 2021) ("Macintosh Networking")
1017	"Success of Motions to Stay Rising, But Why?" (https://www.sternekessler.com/sites/default/files/2020-03/success_of_motions_to_stay_rising_but_why.pdf, last accessed Apr. 13, 2021)
1018	Declaration of Sylvia D. Hall-Ellis, Ph.D.
1019	DocketNavigator listing of cases involving U.S. Patent No. 9,852,004
1020	Swarm Technology LLC's Answer to First Amended Complaint and Counterclaims for Patent Infringement Against Juniper Networks, Inc. and Apstra, Inc., D.I. 64, *Juniper Networks, Inc. v. Swarm Technology LLC*, No. 5:20-cv-03137 (N.D. Cal. Jan. 4, 2022)
1021	Order Re Motion to Dismiss, D.I. 59, *Juniper Networks, Inc. v. Swarm Technology LLC*, No. 5:20-cv-03137 (N.D. Cal. Dec. 21, 2021)
1022	Juniper Networks Inc.'s Notice of Motion to Amend Complaint; Memorandum of Points and Authorities in Support Thereof, D.I. 36, *Juniper Networks, Inc. v. Swarm Technology LLC*, No. 5:20-cv-03137 (N.D. Cal. Feb. 22, 2021)
1023	Corrected Notice of Related Administrative Proceeding at 53, *Juniper Networks, Inc. v. Swarm Technology LLC*, No. 3:20-cv-03137 (N.D. Cal. Sept. 16, 2021)
1024	Screenshot of E2E System (last accessed July 12, 2022)
1025	"Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding" 84 Fed. Reg. 16,654 (Apr. 22, 2019)
1026	*Juniper Networks Inc. v. Swarm Technology LLC*, No. 3:20-cv-3137-JD, 2022 WL 3031211 (N.D. Cal. Aug. 1, 2022)

Exhibit	Description
1027	*Swarm Technology LLC v. Amazon.com, Inc.*, No. 2-21-cv-00438, D.I. 64 (D. Az. Sept. 20, 2021)
1028	Deposition of Dr. Brent E. Nelson Deposition Transcript (July 29, 2022)
1029	Prosecution History for U.S. Patent No. 9,146,777 ("'777 History")
1030	U.S. Patent No. 9,146,777 ("'777 Patent")
1031	Reply Declaration of Dr. Jon B. Weissman ("Weissman Suppl.")

CERTIFICATE OF COMPLIANCE

Per 37 C.F.R. §42.24(c) and (d), the undersigned hereby certifies that the Petition complies with the type-volume limitation of 37 C.F.R. §42.24(c)(i) because, exclusive of exempted portions, it contains 5,592 words as counted by the word-processing program used to prepare it.

Dated: August 24, 2022

Respectfully submitted,

/Joseph F. Edell/

Joseph F. Edell (Reg. No. 67,625)
FISCH SIGLER LLP
5301 Wisconsin Avenue NW
Suite 400
Washington, DC 20015
Phone: 202.362.3524
Email: Joe.Edell.IPR@fischllp.com

CERTIFICATE OF SERVICE

The undersigned certifies that a copy of the foregoing PETITIONER'S REPLY and supporting materials were served via electronic mail to the following attorneys of record for Patent Owner:

Daniel R. Pote
dpote@jsslaw.com
Jennings Strouss & Salmon PLC
One East Washington Street
Suite 1900
Phoenix, Arizona 85004

Michael Kelly
mkelly@beusgilbert.com
Daniel Anderson
danderson@beusgilbert.com
Beus Gilbert McGroder PLLC
701 North 44th Street
Phoenix, Arizona 85008

Dated: August 24, 2022

Respectfully submitted,

/Joseph F. Edell/
Joseph F. Edell (Reg. No. 67,625)
FISCH SIGLER LLP
5301 Wisconsin Avenue NW
Suite 400
Washington, DC 20015
Phone: 202.362.3524
Email: Joe.Edell.IPR@fischllp.com