

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

JUNIPER NETWORKS, INC.,
Petitioner,

v.

FINJAN, INC.,
Patent Owner.

IPR2019-00026
Patent 6,154,844

Before ZHENYU YANG, CHARLES J. BOUDREAU, and
SHEILA F. McSHANE, *Administrative Patent Judges*.

BOUDREAU, *Administrative Patent Judge*.

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

I. INTRODUCTION

A. Background and Summary

Juniper Networks, Inc. (“Petitioner”) filed a Petition (Paper 2, “Pet.”) requesting *inter partes* review of claims 1, 15, and 41 of U.S. Patent No. 6,154,844 (Ex. 1001, “the ’844 patent”). Pet. 1. Finjan, Inc. (“Patent Owner”) filed a Preliminary Response. Paper 6. We granted Petitioner’s request and instituted an *inter partes* review of claims 1, 15, and 41. Paper 7 (“Dec. on Inst.”).

Following institution, Patent Owner filed a Response (Paper 13, “PO Resp.”), Petitioner filed a Reply (Paper 26,¹ “Reply”), and Patent Owner filed a Sur-reply (Paper 29, “Sur-reply”). Petitioner and Patent Owner each filed a Motion to Exclude Evidence, an Opposition to the other’s Motion to Exclude, and a Reply in support of its own Motion to Exclude. Papers 34 (“Pet. Mot. Exclude”), 35 (“PO Mot. Exclude”), 38 (“PO Opp. Mot. Exclude”), 39 (“Pet. Opp. Mot. Exclude”), 40 (“Pet. Reply Mot. Exclude”), 41 (“PO Reply Mot. Exclude”). With the Board’s authorization, Patent Owner also filed a Motion to Terminate (Paper 20), in response to which Petitioner filed an Opposition (Paper 22). Petitioner also filed a Motion to File Under Seal and Enter Proposed Protective Order. Paper 27 (“Mot. to Seal”). A hearing was held on January 10, 2020, and a transcript of the hearing is included in the record as Paper 45 (“Tr.”).

¹ As noted in Section II.H. below, Petitioner’s Reply filed as Paper 26 includes portions that have been designated as confidential and was, accordingly, filed as “Parties and Board Only.” Petitioner also filed a redacted version of its Reply, as Paper 25.

We have jurisdiction under 35 U.S.C. § 6. This Final Written Decision is entered pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons discussed below, we conclude that Petitioner has not shown by a preponderance of the evidence that claims 1, 15, and 41 are unpatentable.

B. Related Matters

The parties report that the '844 patent is the subject of ongoing district court litigation in *Finjan, Inc. v. Juniper Networks, Inc.*, No. 3:17-cv-05659 (N.D. Cal.); *Finjan, Inc. v. ESET, LLC*, No. 3:17-cv-00183 (S.D. Cal.); *Finjan v. Check Point Software Technologies, Inc.*, No. 3:18-cv-02621 (N.D. Cal.); *Finjan, Inc. v. Cisco Systems, Inc.*, No. 5:17-cv-00072 (N.D. Cal.); and *Finjan, Inc. v. SonicWall, Inc.*, No. 5:17-cv-04467 (N.D. Cal.); and that the '844 patent also was asserted in several prior district court litigations. Pet. 1–2; Paper 4, 1. Certain claims of the '844 patent were challenged previously in petitions for *inter partes* review filed by Symantec Corp., Blue Coat Systems, Inc., and Cisco Systems, Inc. Symantec's and Cisco's petitions both were denied, and Blue Coat's petition, which included a motion for joinder with the review requested by Symantec, was dismissed at Blue Coat's request following the denial of Symantec's petition. *Symantec Corp. v. Finjan, Inc.*, Case IPR2015-01894 (PTAB Mar. 11, 2016) (Paper 7) ("1894 Decision" or "1894 Dec."); *Blue Coat Sys., Inc. v. Finjan, Inc.*, Case IPR2016-00498 (PTAB June 20, 2016) (Paper 10); *Cisco Sys., Inc. v. Finjan, Inc.*, Case IPR2017-02154 (PTAB Apr. 3, 2018) (Paper 8) ("2154 Decision" or "2154 Dec.").

C. The '844 Patent

The '844 patent, titled “System and Method for Attaching a Downloadable Security Profile to a Downloadable,” issued November 28, 2000, from U.S. Patent Application No. 08/995,648 (“the '648 application”), filed December 22, 1997. Ex. 1001, codes (21), (22), (45), (54).

The '844 patent is directed to systems and methods “for attaching a Downloadable security profile to a Downloadable to facilitate the protection of computers and networks from a hostile Downloadable.” Ex. 1001, 1:23–27. “A Downloadable is an executable application program . . . downloaded from a source computer and run on [a] destination computer.” *Id.* at 1:44–47. “Examples of Downloadables include Java™ applets . . . , JavaScript™ scripts . . . , ActiveX™ controls . . . , and Visual Basic [scripts].” *Id.* at 1:49–55. “Downloadables may also include plugins, which add to the functionality of an already existing application program.” *Id.* at 1:56–57.

According to the '844 patent, “the Internet has become a major source of many system damaging and system fatal application programs, commonly referred to as ‘viruses,’” and “programmers continue to design computer and computer network security systems for blocking these viruses.” Ex. 1001, 1:32–39. Although “[o]n the most part, these security systems have been relatively successful,” the '844 patent explains, “these security systems are not configured to recognize computer viruses which have been attached to or configured as Downloadable application programs, commonly referred to as ‘Downloadables.’” *Id.* at 1:40–44. The subject matter of the '844 patent, accordingly, “provides systems for protecting a network from suspicious Downloadables.” *Id.* at 1:62–63. “The network system includes an inspector for linking Downloadable security profiles to a Downloadable, and

a protection engine for examining the Downloadable and Downloadable security profiles to determine whether or not to trust the Downloadable security profiles.” *Id.* at 1:65–2:2. In particular, according to the ’844 patent, the system and method “may examine the Downloadable code to determine whether the code contains any suspicious operations, and thus may allow or block the Downloadable accordingly.” *Id.* at 2:65–3:2. Further, “because the system and method . . . link a verifiable Downloadable security profile to a Downloadable, the system and method may avoid decomposing the Downloadable into the Downloadable security profile on the fly.” *Id.* at 3:3–7.

In the operation of a preferred embodiment of the ’844 patent, a developer obtains or generates an uninspected Downloadable and transmits the Downloadable to the inspector for “hostility inspection,” along with a developer certificate used to authenticate the developer. Ex. 1001, 3:55–65. The inspector includes a “content inspection engine” for examining a received Downloadable (e.g., the signed Downloadable from the developer), for generating a Downloadable security profile (“DSP”) based on a rules base, and for attaching the DSP to the Downloadable. *Id.* at 3:66–4:4. The DSP “preferably includes a list of all potentially hostile or suspicious computer operations that may be attempted by the Downloadable, and may also include the respective arguments of these operations.” *Id.* at 4:4–7. Preferably, the content inspection engine performs a full-content inspection, and generating a DSP includes searching the Downloadable code for undesirable patterns and patterns suggesting the code was written by a hacker, as well as comparing a Downloadable against Downloadables known to be hostile, Downloadables known to be non-hostile, and

Downloadables previously examined by the content inspection engine. *Id.* at 4:7–17. After performing content inspection, the inspector attaches an inspector certificate to the Downloadable, verifying the authenticity of the DSP attached to the Downloadable, and transmits the signed, inspected Downloadable to a web server. *Id.* at 4:65–5:5. The web server then may transmit the Downloadable via a network gateway to a computer client. *Id.* at 5:11–13.

D. Illustrative Claim

All three challenged claims are independent. Claims 1 is illustrative and is reproduced below:

1. A method comprising:
receiving by an inspector a Downloadable;
generating by the inspector a first Downloadable security profile that identifies suspicious code in the received Downloadable; and
linking by the inspector the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients.

Ex. 1001, 11:13–20. Independent claims 15 (inspector system) and 41 (computer readable storage medium) include similar limitations. *Id.* at 11:62–12:2, 14:8–18.

E. Prior Art and Asserted Grounds

Petitioner asserted the following grounds of unpatentability in the Petition:

Claim(s) Challenged	35 U.S.C. §	Reference(s)
1, 15, 41	103(a)	Abadi ²
15	103(a)	Abadi, Gryaznov ³

Pet. 3–4, 32–61.

F. Real Parties in Interest

Petitioner identifies Juniper Networks, Inc. as the real party in interest. Pet. 1. Patent Owner identifies Finjan, Inc. as the real party in interest. Paper 4, 1.

II. ANALYSIS

A. Legal Standards

A patent claim is unpatentable under 35 U.S.C. § 103(a) 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;

² Abadi et al., U.S. Patent No. 6,253,370 B1, issued June 26, 2001 (filed Dec. 1, 1997) (Ex. 1008, “Abadi”).

³ Dmitry O. Gryaznov, *Scanners of the Year 2000: Heuristics*, Virus Bull. Conf. 225 (Sept. 1995) (Ex. 1009, 2–11, “Gryaznov”).

(3) the level of skill in the art; and (4) when in evidence, objective evidence of nonobviousness, i.e., secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). “To satisfy its burden of proving obviousness, a petitioner cannot employ mere conclusory statements. The petitioner must instead articulate specific reasoning, based on evidence of record, to support the legal conclusion of obviousness.” *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016). We analyze the asserted grounds with the principles stated above in mind.

B. Claim Construction

In an *inter partes* review, we construe claims of an expired patent according to the standard applied by the district courts.⁴ *See In re Rambus Inc.*, 694 F.3d 42, 46 (Fed. Cir. 2012). Specifically, we apply the principles set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–17 (Fed. Cir. 2005) (en banc). Under that standard, the words of a claim are generally given their “ordinary and customary meaning,” which is the meaning the term would have to a person of ordinary skill at the time of the invention, in the context of the entire patent including the specification. *See Phillips*, 415 F.3d at 1312–13. “In determining the meaning of [a] disputed claim

⁴ We also apply this standard in all *inter partes* reviews based on petitions filed on or after November 13, 2018. *See* Changes to the Claim Construction Standard for Interpreting Claims in Trial Proceedings Before the Patent Trial and Appeal Board, 83 Fed. Reg. 51340 (Oct. 11, 2018) (amending 37 C.F.R. § 42.100(b) effective November 13, 2018). As discussed in greater detail in Section II.C. below, the ’844 patent claims the benefit of, *inter alia*, U.S. Patent Application No. 08/790,097, filed January 29, 1997 (“the ’097 application”). Based on the filing date of that application, the ’844 patent expired no later than January 29, 2017. *See* 35 U.S.C. § 154(a)(2). The parties concur that the ’844 patent has expired. Pet. 16; Prelim. Resp. 4.

limitation, we look principally to the intrinsic evidence of record, examining the claim language itself, the written description, and the prosecution history, if in evidence.” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 469 F.3d 1005, 1014 (Fed. Cir. 2006) (citing *Phillips*, 415 F.3d at 1312–17). Extrinsic evidence is “less significant than the intrinsic record in determining ‘the legally operative meaning of claim language.’” *Phillips*, 415 F.3d at 1317. Only those terms in controversy need to be construed, and only to the extent necessary to resolve the controversy. *See Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

As we explained in our Decision on Institution, Petitioner contends in the Petition that each claim term should be given its plain and ordinary meaning in this proceeding and that “no specific construction of any claim term is required because the prior art relied upon meets each of the limitations under any reasonable construction,” but nonetheless addresses several terms in light of arguments made by Patent Owner in previous proceedings. Dec. on Inst. 7 (citing Pet. 17–27). We preliminarily construed each of those terms in the Decision on Institution, based on the record then before us. Dec. on Inst. 7–15. Patent Owner responds to each of Petitioner’s proposed constructions and our preliminary constructions in its Response; Petitioner responds to Patent Owner’s arguments in its Reply; and Patent Owner further responds in its Sur-reply. PO Resp. 7–14; Reply 1–11; Sur-reply 1–7. We address the parties’ arguments with respect to each term in turn below.

1. “Downloadable”

The term “Downloadable” is recited in each of the challenged claims. In our Decision on Institution, we preliminarily adopted Petitioner’s

unopposed proposed construction of this term as “an executable application program, which is downloaded from a source computer and run on the destination computer.” Dec. on Inst. 7 (citing Pet. 17–18; Prelim. Resp. 5).⁵

Notwithstanding Patent Owner’s express agreement with the adopted construction in its Response (PO Resp. 7), Petitioner contends in its Reply that “Patent Owner implicitly proposes two incorrect meta-constructions” in its Response by arguing, “[f]irst, . . . that ‘executable application program’ should be interpreted to exclude intermediate code—*i.e.*, code that requires interpreting or compiling before execution,” and “[s]econd, . . . that the Downloadable cannot be modified in any way during transit from the source computer to the destination computer, *e.g.* by attaching annotations.” Reply 2 (citing PO Resp. 38, 40, 42–43). Petitioner argues that the Board should reject Patent Owner’s positions as being “contradicted by the ’844 Patent’s specification” and “untenable in view of [Patent Owner’s] prior admissions.” *Id.* Specifically, Petitioner argues, the ’844 patent states that “[e]xamples of Downloadables include Java™ applets . . . JavaScript™ scripts . . . and Visual Basic,” which, Petitioner contends, a person of ordinary skill in the art would recognize to be “types of intermediate (and ‘executable’) code in that they cannot be directly executed by a CPU but instead must be interpreted . . . or compiled.” *Id.* (citing Ex. 1001, 1:48–55). Petitioner further argues that the ’844 patent states that “the Downloadable security profile can be ‘linked to the Downloadable via attachment’, and

⁵ We also previously adopted essentially the same construction in IPR2017-02154. *See* 2154 Dec. 13 (adopting agreed-upon construction of “Downloadable” as “an executable application program that is downloaded from a source computer and run on the destination computer”).

Patent Owner has admitted that the mechanics of “attaching [code] to the applet being sent in the data packet” means that applet’s executable code is altered.” *Id.* at 2–3 (alteration in original) (emphases omitted) (footnote omitted) (citing Ex. 1001, claim 2; Ex. 1040, 40; Ex. 1041 ¶ 74). Petitioner contends that “because attaching a security profile to an applet is a modification and the ’844 Patent expressly contemplates Downloadables with attached security profiles, the claims of the ’844 Patent must be interpreted broadly enough to encompass such modifications.” *Id.* at 3. Furthermore, Petitioner asserts, “there is nothing in the agreed construction that precludes modification of the executable application program during transit.” *Id.*

Patent Owner responds in its Sur-reply that “Petitioner’s arguments are premised on a far broader construction” than the parties have agreed to, “one that focuses simply on whether the code is executable, thereby ignoring the fact that the executable application program that is downloaded from a source computer must be the same executable application program run on the destination computer.” Sur-reply 1–2 (emphasis omitted). According to Patent Owner, whereas Petitioner focuses on the attachment of a DSP to a Downloadable in the claimed method in support of its argument that modification to the source code of the Downloadable is allowable, the ’844 patent explains that “a DSP is attached to a ‘signed’ Downloadable or a Downloadable with a Downloadable ID, meaning that the Downloadable has a unique identifier that shows the Downloadable has not been modified in any way.” *Id.* at 2 (citing Reply 2–3; Ex. 1001, 8:65–67, Fig. 2). Accordingly, Patent Owner contends, a person of ordinary skill in the art would thus understand that “the ‘attaching’ that occurs in the ’844 Patent

does not alter the unique ‘signed’/identifier/hash value of Downloadable because there is no modification of the Downloadable code,” and “[t]hus, Petitioner’s interpretation is inconsistent with what the ’844 Patent requires.” *Id.* at 2–3.

Having considered the parties’ arguments, we see no reason to disturb their agreed construction of “Downloadable” as “an executable application program, which is downloaded from a source computer and run on the destination computer,” with which the parties state their express agreement. We clarify in Section II.D.2. below that we agree with Patent Owner that the executable application program that is to be run on the destination computer must be the same executable application program that is downloaded from the source computer within the agreed construction. Sur-reply 1–2. This interpretation is not a question of the construction of the term “Downloadable” itself, but rather flows from the express language of the challenged claims, in which “the Downloadable” that is “ma[de] . . . available to web clients” in the final step of each of independent claims 1 and 41 has its antecedent basis in the first claim step, “receiving . . . a Downloadable.” Likewise, the antecedent in claim 15 for “the Downloadable” that is made available to web clients is the Downloadable in which suspicious code is identified by the content inspection engine. We also agree with Patent Owner that the claimed “linking” of a DSP “to the Downloadable” does not compel a different understanding because we are not persuaded that the linking contemplated in the ’844 patent requires modification of the code of the Downloadable itself. Sur-reply 2–3.

2. *“Downloadable security profile that identifies suspicious code in [the received / a] Downloadable”*

Claims 1 and 41 each recite a “Downloadable security profile that identifies suspicious code in the received Downloadable.” Ex. 1001, 11:15–17, 14:13–14. Claim 15 similarly recites a “Downloadable security profile that identifies suspicious code in a Downloadable.” *Id.* at 11:65–66. In our Decision on Institution, after considering the arguments set forth in the Petition and Preliminary Response, we preliminarily adopted Petitioner’s proposed construction of “Downloadable security profile” as “information that describes characteristics of a Downloadable for use in protecting a computer system.” Dec. on Inst. 8 (citing Pet. 18–19; Prelim. Resp. 5). We also preliminarily construed the phrase “identifies suspicious code in the received Downloadable,” as requiring that the Downloadable security profile “must at least include, but is not necessarily limited only to, an identification of hostile, potentially hostile, undesirable, or potentially undesirable code in the corresponding Downloadable,” based in part on disclosure in the ’388 application, incorporated by reference in the ’844 patent, that “[i]t is to be understood that the term ‘suspicious’ includes hostile, potentially hostile, undesirable, potentially undesirable, etc.” Dec. on Inst. 8–11 (emphases omitted) (citing Ex. 1007, 7:15–16; Pet. 19–21; Prelim. Resp. 5–6).

In its Response, Patent Owner agrees with our preliminary construction as to the phrase “identifies suspicious code in [the received / a] Downloadable,” but disagrees with respect to “Downloadable security profile,” contending that the complete phrase, “Downloadable security profile that identifies suspicious code in [the received / a] Downloadable,” should be construed as “a profile that must at least include, but is not necessarily limited only to, an identification of hostile, potentially hostile,

undesirable, or potentially undesirable code in the corresponding Downloadable.” PO Resp. 7–8. According to Patent Owner, the plain and ordinary meaning of “Downloadable security profile” is simply “profile,” and “[b]y rephrasing the term, Petitioner is reading the word ‘profile’ out of the term.” *Id.* at 8. Patent Owner further contends Petitioner’s proposed construction “is unnecessary and improperly imports limitations that is [sic] not necessary to understand the meaning and scope of the claim term.” *Id.* at 8–9.

In its Reply, Petitioner contends that, although Patent Owner disputes the construction of “Downloadable security profile,” Patent Owner “does not argue that the asserted prior art fails to teach a ‘Downloadable security profile’ under any proposed construction,” and “[t]herefore, any remaining dispute regarding this term is irrelevant.” Reply 3–4. Patent Owner maintains its disagreement in its Sur-reply, noting that Petitioner does not dispute or disagree with Patent Owner’s proposed construction in its Reply. Sur-reply 3.

Having considered the full record developed during trial, we agree with Patent Owner that the replacement of the word “profile” with “information” does not add any clarity to the meaning of this phrase and that the proper construction of “Downloadable security profile that identifies suspicious code in [the received / a] Downloadable” in the context of the ’844 patent is “profile that must at least include, but is not necessarily limited only to, an identification of hostile, potentially hostile, undesirable, or potentially undesirable code in the corresponding Downloadable.” PO Resp. 7–8. Regardless, we do not discern that the difference between the

parties' positions with respect to these phrases is dispositive of any issues in this proceeding.

3. *“linking [by the inspector] the first Downloadable security profile to the Downloadable”*

Petitioner proposed in the Petition that the term “linking” should be construed “to include any type of association,” such that the phrase “linking . . . the first Downloadable security profile to the Downloadable” in each of the challenged claims should be construed as “associating the first Downloadable security profile with the Downloadable.” Pet. 22. In support of that construction, Petitioner cited disclosure in the '844 patent that “[t]he term ‘linking’ herein will be used to indicate an association between the Downloadable 205 and the DSP 215 (including using a pointer from the Downloadable 195 to the DSP 215, attaching the DSP 215 to the Downloadable 205, etc.).” *Id.* (quoting Ex. 1001, 6:20–24). After considering Petitioner’s arguments and cited evidence, as well as Patent Owner’s argument in its Preliminary Response that “‘linking...’ should be given its plain and ordinary meaning” (Prelim. Resp. 8), we adopted Petitioner’s construction in the Decision on Institution (Dec. on Inst. 11–12). In its Response, Patent Owner agrees that the adopted construction is the plain and ordinary meaning of the recited phrase. PO Resp. 9. In view of the full record and the parties’ agreement, we do not alter our prior construction.

4. *“web server”*

The term “web server” is recited in each of the challenged claims. Ex. 1001, 11:19, 12:1, 14:17. Petitioner argued in the Petition that the term “web server” should be construed as “a computer that stores content that is

available to other computers to download over an external network,” which Petitioner argues is consistent with the ’844 patent’s use of the term to refer broadly to a computer located in an external network that stores content and makes that content available to other computers. Pet. 22–25 (citing Ex. 1001, 3:42–43, 5:3–6, 9:10–16, 10:24–32, 10:56–58, Fig. 1).

After considering Petitioner’s arguments and cited evidence, as well as Patent Owner’s arguments in its Preliminary Response that, for example, “web server” is a well-understood term in the art that does not need construction and that “Petitioner’s construction . . . is inconsistent with the intrinsic record and well-understood meaning of the term” (Prelim. Resp. 9), we agreed with Patent Owner in our Decision on Institution that there was no need shown for express construction and that the plain and ordinary meaning of “web server” should apply. Dec. on Inst. 12–13. We noted in our Decision, however, that the only evidence of record with regard to such plain and ordinary meaning was an undated “Dictionary of IBM & Computing Terminology” cited by Patent Owner,⁶ and that, contrary to Patent Owner’s suggestion that that dictionary defines “web server” as “a web server that can serve web pages” (Prelim. Resp. 9), the definition of “Web server” therein is, instead, “[a] server that is connected to the Internet and is dedicated to serving Web pages” (Ex. 3001, 93). Dec. on Inst. 13. We further explained that “Internet” is defined there as “[t]he worldwide

⁶ See Dictionary of IBM & Computing Terminology, <https://www.ibm.com/ibm/history/documents/pdf/glossary.pdf>. Patent Owner provided only the foregoing URL for the cited dictionary and did not enter a copy of the dictionary as an exhibit in this case. Along with our Decision on Institution, we entered as Exhibit 3001 in this proceeding a copy of the dictionary definition as it existed at the cited URL as of April 9, 2019.

collection of interconnected networks that use the Internet suite of protocols and permit public access” (Ex. 3001, 45) and that “Web page” is defined as “[a]ny document that can be accessed by a uniform resource locator (URL) on the World Wide Web” (*id.* at 93). Dec. on Inst. 13. Accordingly, we stated that we understood that “the plain and ordinary meaning of ‘web server’ requires at least a server connected to the Internet and capable of serving documents accessible by a URL on the World Wide Web.” *Id.*

In its Response, Patent Owner essentially repeats its argument from its Preliminary Response, asserting that the plain and ordinary meaning of “web server” is “a server that can serve web pages.” PO Resp. 13 (citing Ex. 2016 (Declaration of Dr. Trent Jaeger) ¶¶ 51–52). Patent Owner contends that the Board’s understanding of the plain and ordinary meaning “unnecessarily requires a connection to the Internet,” as “web pages can be served on an internal network that does not require the Internet.” *Id.* at 13–14 (citing Ex. 2016 ¶ 51; Ex. 2040 (Nielson Deposition Transcript), 39:21–40:6). Petitioner argues in its Reply that although Patent Owner disputes the Board’s interpretation, Patent Owner does not argue that the asserted prior art does not teach a web server, and accordingly, that the Board need not revisit its prior construction. Reply 4 (citing PO Resp. 37–56; Ex. 1037 (Jaeger Deposition Transcript), 118:14–20). In its Sur-reply, Patent Owner repeats once again its argument that the plain and ordinary meaning of “web server” is “a server that can serve web pages,” and also argues that “Petitioner does not dispute or disagree with Patent Owner’s proposed construction.” Sur-reply 7.

In view of the parties’ arguments, we are not persuaded to alter our conclusion in the Decision on Institution that the plain and ordinary meaning

of a web server requires a server connected to the Internet and capable of serving documents accessible by a URL on the World Wide Web. Dec. on Inst. 13 (citing Ex. 3001, 45, 93). As we explained in that decision, this conclusion is based on the definitions of “Web server” and “Web page” in the technical dictionary provided by Patent Owner in its Preliminary Response. *See id.*; *see also* Prelim. Resp. 9 (citing Dictionary of IBM & Computing Terminology, <https://www.ibm.com/ibm/history/documents/pdf/glossary.pdf>). This interpretation is fully consistent with the usage of the term web server in the ’844 patent, which depicts external computer network 105 interposed between web server 185 and web client 135 (Ex. 1001, Fig. 1) and states, for example, that “web server 185 [is] . . . coupled to . . . external computer network 105” (*id.* at 3:39–44), that “web server 185 . . . *transmits the Downloadable via the network gateway 110* to the computer client 130” (*id.* at 5:11–13 (emphasis added)), and that “communications interface 825 [of web server 185] couples the signal bus 825 to the external computer network 105” (*id.* at 10:33–34). Although, as Patent Owner points out, the ’844 patent states that “[o]ne skilled in the art will recognize that connections to external or internal network systems are merely exemplary, and alternative embodiments may have other connections” (*id.* at 3:44–47), we do not find sufficient support in that generic statement for broadening “web server” beyond its ordinary meaning to, in effect, read out the word “web.” And indeed, notwithstanding Patent Owner’s assertion that “web pages can be served on an internal network,” the testimony of Dr. Jaeger cited in support of that assertion in Patent’s Owner’s Response and Sur-reply states, in part, that “the ’844 Patent limited web server such that it *must be* on the Internet.” Ex. 2016 ¶ 52.

5. *linking the first Downloadable security profile to the Downloadable “before a web server makes the Downloadable available to web clients”*

Each of the challenged claims recites linking the first Downloadable security profile to the Downloadable “before a web server makes the Downloadable available to web clients.” Ex. 1001, 11:18–20, 11:67–12:2, 14:16–18. Petitioner argued in the Petition that this phrase does not need to be construed and should be given its plain and ordinary meaning, but that Patent Owner had previously taken positions in co-pending litigation that were inconsistent with the plain language and the prosecution history, including by arguing that “before a web server makes the Downloadable available to web clients” means “any time before the particular clients are permitted to access the Downloadable” and that “linking” can occur “anywhere in the network, including on a network gateway.” Pet. 25 (citing Ex. 1010, 10). In the Decision on Institution, after taking into account the arguments in the Petition and Patent Owner’s Preliminary Response, we agreed with Petitioner that no express claim construction is required for the recited claim phrase. Dec. on Inst. 13–15 (citing Pet. 25; Prelim. Resp. 12–14). We further explained, however, that we were not persuaded by an argument by Patent Owner that inspection and linking at a network gateway necessarily could satisfy this claim limitation. *Id.* at 15 (citing Prelim. Resp. 12–14). In particular, we explained that we could find no evidence that such can be considered a “preferred” embodiment, as Patent Owner had alleged. *Id.* We also explained that we agreed with Petitioner that the claims must be construed in light of the May 18, 2000, amendment of each of then-pending independent claims 1, 15, and 41 of the ’648 application to add the phrase “before a web server makes the Downloadable available to web clients,”

accompanied by remarks distinguishing the claims thus amended from the Ji reference cited by the Examiner. *Id.* (citing Ex. 1002, 201–205; Pet. 25–26).

In its Response, Patent Owner makes essentially the same arguments as in its Preliminary Response, maintaining that “the plain and ordinary meaning of this term is that the ‘linking’ of a Downloadable security profile to the Downloadable occurs before web clients can access the Downloadable from a web server.” PO Resp. 10–12. Patent Owner disputes that it disclaimed embodiments in which the inspector is located on a gateway, contending that the Ji reference cited by the Examiner during prosecution was distinguished on the basis that the method described in the ’844 patent is less resource intensive than Ji because it can reuse known security profiles, thus saving processing time and reducing the load on a gateway. *Id.* at 11–12 (citing Ex. 1002, 204). Petitioner addresses these arguments in its Reply, and Patent Owner responds in its Sur-reply. Reply 4–11; Sur-reply 4–6.

Having considered the full record, we conclude that there is no need to construe this term, including whether or not the recited “linking” may occur “at [a] network gateway,” to resolve the controversy before us, and we maintain our prior determination that no express claim construction is required. Dec. on Inst. 15. The application of the ordinary meaning of this claim language to the issues in this case is discussed in Sections II.C.3 and II.D.2.c below.

C. Priority Date of the ’844 Patent

In our Decision on Institution, we explained that the parties disputed the priority date of the challenged claims and, thus, whether Abadi is properly considered as prior art. Dec. on Inst. 15 (citing Pet. 10–16; Prelim. Resp. 15–24). Following consideration of the parties’ respective arguments,

we concluded that the challenged claims were not entitled to a priority date any earlier than the December 22, 1997, filing date of the application for the '844 patent, and accordingly, we also concluded that Abadi, which issued from an application filed December 1, 1997, is prior art. *Id.* at 15–24. The parties raised additional arguments regarding Patent Owner's priority claim in their post-institution briefing. PO Resp. 14–26; Reply 11–16; Sur-reply 7–13. Having considered those additional arguments, we remain unpersuaded, for the reasons that follow, that the challenged claims are entitled to any earlier priority date.

1. Background

The Related U.S. Application Data field on the front page of the '844 patent, as issued, included a reference to Provisional Application No. 60/030,639, filed November 8, 1996 (“the '639 provisional”; Ex. 1005⁷). Ex. 1001, code (60). Separately, in a section titled “PRIORITY REFERENCE TO RELATED APPLICATIONS,” the '844 patent states that the '648 application “claims benefit of and . . . incorporates by reference [the '639 provisional]; patent application Ser. No. 08/964,388, . . . filed on Nov. 6, 1997 . . . ; and patent application Ser. No. 08/790,097, . . . filed on Jan. 29, 1997.” *Id.* at 1:7–17. That statement, however, does not indicate how the '648 application and the earlier applications are related to one another (e.g., as continuations, divisionals, or continuations-in-part). *See* 37 C.F.R. § 1.78(a)(2) (1997) (“Any nonprovisional application claiming the benefit of one or more prior filed copending nonprovisional applications . . .

⁷ An additional copy of the '639 provisional, with an attached “Appendix” and several additional bibliographic pages, is filed as Exhibit 2015. *See* PO Resp. 18 n.3.

must contain or be amended to contain in the first sentence of the specification following the title a reference to each such prior application . . . indicating the relationship of the applications.”). On March 18, 2016, Patent Owner filed a Request for Certificate of Correction, requesting that the reference to the ’639 provisional be deleted from the front page of the ’844 patent; that the following be inserted in its place:

- (60) This application is a continuation-in-part of U.S. Patent Application No. 08/964,388, filed on Nov. 6, 1997, now U.S. Patent No. 6,092,194 and a continuation-in-part of U.S. Patent Application No. 08/790,097, filed on Jan. 29, 1997, now U.S. Patent No. 6,167,520;

and that the “PRIORITY REFERENCE TO RELATED APPLICATIONS” be amended in essentially four regards: (1) to omit the claim of benefit of the ’639 provisional; (2) to add that the ’648 application “is a continuation-in-part of” the ’388 application and U.S. Patent Application No. 08/790,097 (“the ’097 application”; Ex. 1006, 6–41); (3) to add that the ’388 and ’097 applications are now U.S. Patent Nos. 6,092,194 (“the ’194 patent”; Ex. 2005) and 6,167,520 (“the ’520 patent”; Ex. 2006), respectively; and (4) to add that the ’194 and ’520 patents themselves claim benefit to the ’639 provisional. *See* Ex. 2007, 1 (“Request”).

On August 16, 2016, the Office issued a Response to the Request for Certificate of Correction, stating that the Request was improper “[w]ith respect to the alleged error in Col. 1, 1st para”—referring to the requested amendment to the “PRIORITY REFERENCE TO RELATED APPLICATIONS”—because “comparison of the printed patent with the corresponding location in the application file reveals that there is no discrepancy.” Ex. 2008, 2 (“Office Response”). The Office Response

further stated that a certificate of correction would “issue without the inclusion of col. 1, 1st para. as requested” (*id.*), and indeed, the Office issued a Certificate of Correction on September 6, 2016, including the requested corrections to the front page of the patent but not the requested amendment to the “PRIORITY REFERENCE TO RELATED APPLICATIONS” (*see* Ex. 2009, 1).

2. The Parties’ Positions

In the Petition, Petitioner argues that although the ’844 patent indicates that it claims priority from the ’639 provisional and the ’388 and ’097 applications (collectively, the “Earlier Applications”), the Earlier Applications do not contain adequate written description support for the challenged claims. Pet. 10–14. As an initial matter, Petitioner asserts that the Board “has previously found that the ’844 [p]atent is not entitled to an earlier priority date based on” the Earlier Applications. *Id.* at 11. In support of that assertion, Petitioner cites our decision denying institution in IPR2015-01894, in which we explained that the record then before us did not establish that the challenged claims are entitled to the benefit of any filing date earlier than December 22, 1997. *Id.* at 11–12 (citing 1894 Dec. 6). Petitioner also cites our decision denying institution in IPR2017-02154, in which we explained that we did not find support in the ’639 provisional or the ’097 application for linking a Downloadable security profile to a Downloadable “before a web server makes the Downloadable available to web clients,” as recited in each of the challenged claims. *Id.* at 12 (citing 2154 Dec. 12).

Petitioner notes that, although the petitioner in IPR2017-02154 did not contest the applicability of the ’388 application’s priority date to the

'844 patent, that application also fails to provide adequate written description for the claim element of “linking the first downloadable security profile to the Downloadable before the Downloadable is made available to web clients by a web server.” Pet. 12. In particular, Petitioner argues,

As with the '639 Provisional and '097 Application, the '388 Application does not even mention the term “web server.” Nor does it describe or suggest, either explicitly or implicitly, “linking” the DSP to the Downloadable “before the Downloadable is made available to web clients by a web server.” This is unsurprising given that the '388 Application purports to disclose an entirely different invention than the Challenged Claims. Indeed, the '388 Application—as well as the '639 Provisional and '097 Application—are directed to an *internal* network security system for “deriving downloadable security profile data” from Downloadables that a web server has already made available.

Id. at 12–13 (citing Ex. 1005, 6:1–14; Ex. 1006, 7:15–15; Ex. 1007, 7:1–13). Relying on testimony of its declarant, Dr. Seth Nielson, Petitioner further contends:

Given that none of the Earlier Applications contains any discussion of “linking” a DSP with a Downloadable “before a web server makes the Downloadable available to web clients,” a POSITA reading those applications could not reasonably conclude that the inventor had possession of the invention claimed in the Challenged Claims, at the time that the Earlier Applications were filed.

Id. at 14 (citing Ex. 1003 (Nielson Declaration) ¶ 35).

Lastly, Petitioner argues that priority to the Earlier Applications was not properly claimed during prosecution, and, thus, the '844 patent cannot claim priority from the Earlier Applications for this independent reason. *Id.* at 14–16. In support, Petitioner points to (1) the inability of the '844 patent to claim priority directly from the '639 provisional given that more than a

year elapsed between the filing of the '639 provisional and the filing of the '648 application; (2) the failure of the '648 application as filed (and as issued as the '844 patent) to indicate how the '097 and '388 applications are related thereto; and (3) the Office's partial denial of Patent Owner's Request for Certificate of Correction, as detailed in the Background section above.

Id.; see *supra* Section II.C.1.

In summary, Petitioner argues that the challenged claims are entitled to a priority date no earlier than December 22, 1997.

Patent Owner responds that the '844 patent is entitled to the November 8, 1996, filing date of the '639 provisional because the patent meets all requirements under 35 U.S.C. §§ 119 and 120 to claim the benefit of the provisional filing date. PO Resp. 14–15. First, according to Patent Owner, the '844 patent and each of its ancestral applications contain specific references to the earlier-filed applications. In particular, Patent Owner contends, the '844 patent claims the benefit of the '639 provisional through the '388 and '097 applications, which issued as the '194 and '520 patents, respectively, and the '194 and '520 patents themselves claim the benefit of the '639 provisional. *Id.* at 15 (citing Ex. 1001, 1:7–19; Ex. 2005, 1:16–20; Ex. 2006, 1:6–11). Patent Owner also points to the Office's statement in the Office Response that “[w]ith respect to the alleged error in Col. 1, 1st para, comparison of the printed patent with the corresponding location in the application file reveals that there is no discrepancy,” and contends that “[t]he Office . . . found that there was no error in the priority chain as originally recited.” *Id.* at 17 (citing Ex. 1002, 417).

Second, Patent Owner contends, the disclosure of the '639 provisional supports the challenged claims of the '844 patent under 35 U.S.C. § 112, and

the '639 provisional also is incorporated by reference into the '844, '194, and '520 patents, providing continuity of disclosure from the '639 provisional through the '844 patent. PO Resp. 17–18 (citing Ex. 1001, 1:8–12; Ex. 2005, 1:16–20; Ex. 2006, 1:6–11; Ex. 2015). Patent Owner contends that Petitioner's reliance on the Board's decisions on institution in IPR2015-01894 and IPR2017-02154 is misplaced because the necessary conditions for issue preclusion are not met in this case, particularly because the determination of the priority date was not necessary to the Board's institution decisions in those cases and also because Patent Owner did not have an opportunity to litigate the written description question there. *Id.* at 18–19 (citing *Soverain Software LLC v. Victoria's Secret Direct Brand Mgmt., LLC*, 778 F.3d 1311, 1315 (Fed. Cir. 2015); *Symantec Corp. v. Finjan, Inc.*, IPR2015-01894, Paper 7 at 16, 22–23, 28–29 (PTAB Mar. 11, 2016); *Cisco Sys., Inc. v. Finjan, Inc.*, IPR2017-02154, Paper 8 at 33–34 (PTAB Apr. 3, 2018)).

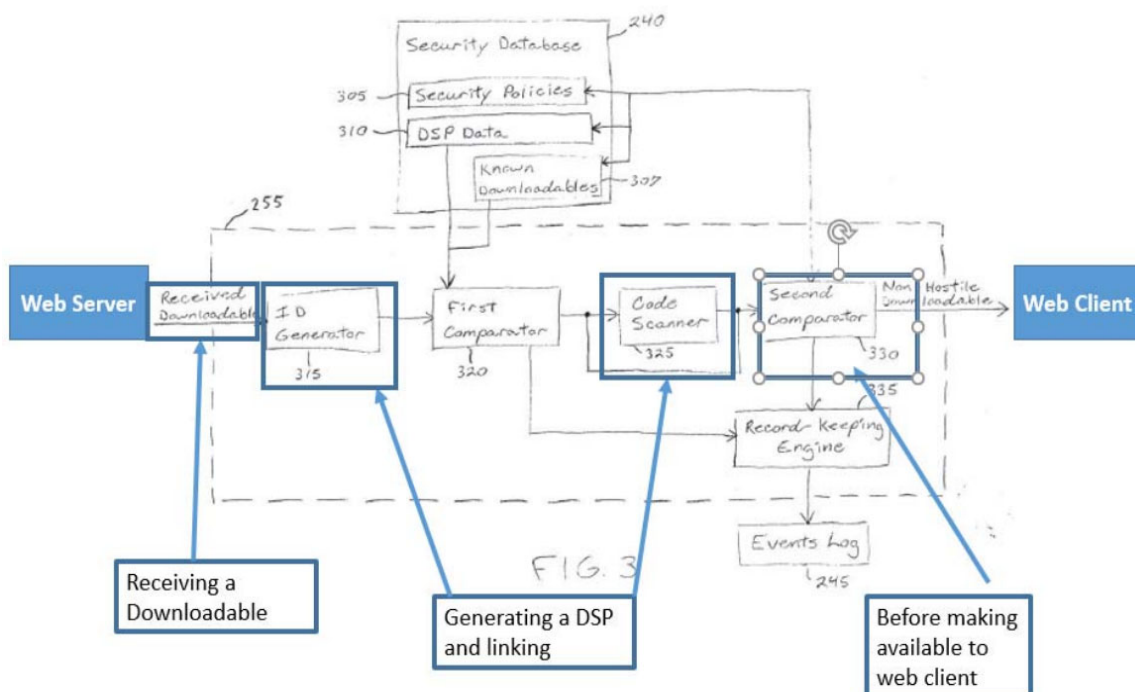
We explained in the Decision on Institution in this proceeding that we were not persuaded that the portions of the '639 provisional cited by Patent Owner in its Preliminary Response demonstrate possession by the inventors of “*linking by the inspector the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients.*” Dec. on Inst. 23. Acknowledging that prior determination in its Response, Patent Owner maintains that the '639 provisional discloses support for that element. PO Resp. 19. In particular, according to Patent Owner:

The '639 Application describes a system which includes “an ID generator 315, a code scanner 325, and a second comparator 330”, among others, for examining Downloadables.

Ex. 2015 ('639 Provisional) at 14. The system is connected to the Internet. Jaeger Decl. at ¶59 (explaining that the external network 105 includes a web server, such as the web server for “cnn.com,” which is connected to the Internet). The system examines a Downloadable received from the external network, analyzes the Downloadable to generate and link a DSP, and then forwards the Downloadable to a web client (or discards it). Jaeger Decl. at ¶59.

Because a DSP is linked to the Downloadable before the Downloadable reaches a web client, the “linking . . . before a web server makes a Downloadable available to a web client” is disclosed in the ’639 Provisional Application. *See Blue Coat Systems, Inc.*, 879 F.3d at 1306 (finding that “the limitation requiring that linking occur before a downloadable is ‘ma[de] . . . available to web clients’ could reasonably be understood to require that linking occur at some point before users are *permitted to access* that downloadable—but not necessarily before the downloadable is made available on the Internet.”) (emphasis added).

PO Resp. 19–20. Patent Owner provides the following annotated version of Figure 3 of the '639 provisional:

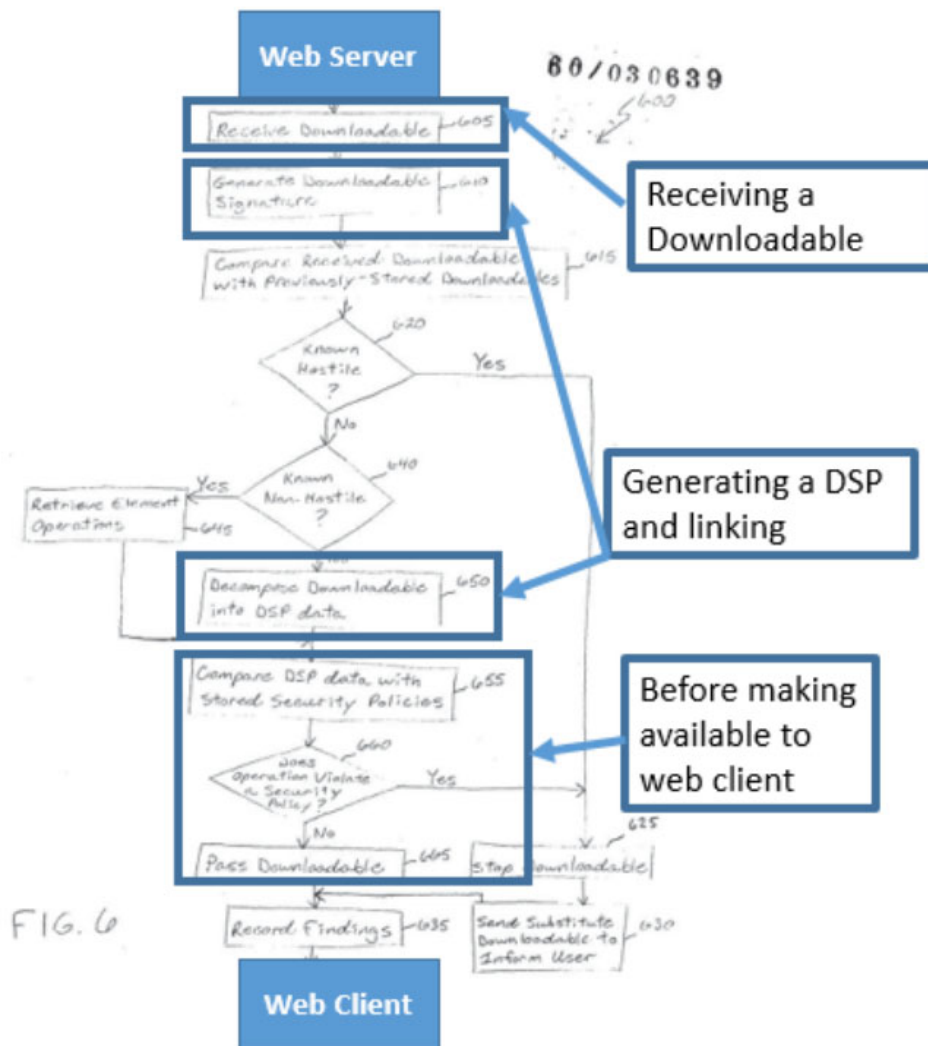


PO Resp. 21. According to Patent Owner, the annotated Figure 3 block diagram above illustrates how the '639 provisional “connects to a web server and a web client.” *Id.* at 20. For example, Patent Owner contends,

ID generator 315 “receives Downloadables from external computer network 105 via external communications interface 210 [and] generates a digital signature for each Downloadable.” Ex. 2015[,] 15. The code scanner 325 generates a DSP by analyzing an unknown Downloadable and “sends the Downloadable and the corresponding DSP data into second comparator 330.” Ex. 2015[,] 16 (stating that the code scanner 325 parses the Downloadable), 17 (stating that the code scanner 325 “decodes and registers the command and the command parameters as DSP data”). *See also* [Ex. 2016] ¶¶ 60-62.

PO Resp. 21. Patent Owner further contends that the '639 provisional “discloses at least two examples for linking, or associating, the DSP to the Downloadable.” *Id.* at 22. First, Patent Owner alleges, “the sending of the Downloadable and the corresponding DSP data from the code scanner to the second comparator shows that the Downloadable and the DSP are associated (linked), such that the second comparator 330 can make a decision on the Downloadable based on the DSP data.” *Id.* (citing Ex. 2015, 16–17; Ex. 2016 ¶ 62). Patent Owner alleges that Petitioner’s expert, Dr. Nielson, confirms this understanding. *Id.* (citing Ex. 2040, 64:10–18). Second, Patent Owner contends, “the DSP and the Downloadable are linked through a Downloadable ID, where the Downloadable ID represents an association between the DSP and the Downloadable.” *Id.* (citing Ex. 2015, 42; Ex. 2016 ¶¶ 59–62). According to Patent Owner, the linking of the DSP to the Downloadable “occurs before a web server makes a Downloadable available to a web client because the system disclosed in the '639 Provisional examines the Downloadable and links it with a security profile prior to the

Downloadable becomes [sic] available at a client computer, such as a computer located in the internal network.” *Id.* at 22–23 (citing Ex. 2015, 37, Fig. 6; Ex. 2016 ¶¶ 61–62; Ex. 2040, 58:2–59:20). Patent Owner contends that “this process is also shown in Figure 6 and the example DSP of the ’639 application.” *Id.* at 23. To illustrate, Patent Owner provides the following annotated version of Figure 6 of the ’639 provisional:



PO Resp. 23. With reference to steps shown in the annotated Figure 6 flowchart above, Patent Owner argues,

As discussed in the ’639 Provisional, a Downloadable is received from a web server and then the inspector will generate

an ID for the Downloadable. Ex. 2015[,] 20 (14:1–6) (“ID generator 315 in step 610 generates a signature representing the received Downloadable.”). If the Downloadable is unknown, the unknown Downloadable is scanned to generate a Downloadable security profile. Ex. 2015, [] 20 (14:1–2, 17–20), 21 (15:1–4). The profile is linked to the Downloadable using the Downloadable ID. Ex. 2015[,] at 42 (identified as the “Applet ID(signature): xxx-xxxxx-xxxx-xxxx”). . . . [A]n example security profile includes the Downloadable ID, thus showing linking at step 650. . . . The linking and generation of the DSP must occur before the inspector can pass on the Downloadable on to the web client because at step 655, the DSP data is compared with a security policy. If policy violates a security policy, the Downloadable is blocked (step 625), otherwise the Downloadable is passed on the web client at step 665. Ex. 2015, [] 21 (15:5–13).

PO Resp. 24–25. Finally, referring to the “Appendix” to the ’639 provisional, Patent Owner contends that written description support for linking is found in the ’639 provisional, including in portions omitted by Petitioner and not reviewed by Petitioner’s expert. *Id.* at 25 (“*Compare* Ex. 2001 [sic]⁸ (including the Appendix at pgs. 32–40) *with* Ex. 1005 (omitting the Appendix)”).

In its Reply, Petitioner raises three principal arguments in response to Patent Owner’s arguments. Reply 11–16.

First, Petitioner argues that Patent Owner is precluded from arguing that the ’639 provisional contains sufficient written description, alleging that “[t]his issue was actually litigated and determined by a valid and final judgment in IPR2017-02154 after the parties had a full and fair opportunity to litigate the issue” and that “[t]he determination was also essential to the

⁸ We understand Patent Owner’s citation to “Ex. 2001” to refer to Exhibit 2015.

judgment” in that case, such that “[a]ll of the elements of issue preclusion are satisfied here.” Reply 11–13.

Second, Petitioner contends that Patent Owner’s substantive arguments that the ’639 provisional contains sufficient written description to support the challenged claims are incorrect under the proper construction of the limitation “before a web server makes the Downloadable available to web clients.” Reply 13–14. According to Petitioner, Patent Owner distinguished Ji during prosecution by amending the claims to include that limitation, support for which can only be found in the new matter added to the ’844 patent. *Id.* at 14. In contrast, Petitioner argues, “[t]he only component in the ’639 Provisional that even arguably generates and links a Downloadable security profile is,” as in Ji, “a gateway.” *Id.*

Finally, Petitioner argues that, even if the earlier applications had contained sufficient written description support, priority still would not attach because Patent Owner failed to properly claim priority under 37 C.F.R. § 1.78(a)(2). Reply 14–15. Petitioner points out that although Patent Owner filed a Petition to Amend Priority Claim to try to fix the defect in its priority claim, the Office denied that petition in relevant part, and the claim of priority has not been fixed. *Id.* at 15. Petitioner contends that Patent Owner mischaracterizes the basis for the Office’s denial of its request by suggesting that the Office found that there was no error in the priority chain as originally recited, whereas the Office instead denied the request as being an improper attempt to perfect a claim of priority. *Id.* at 15–16.

Patent Owner addresses Petitioner’s arguments in its Sur-reply, contending, *inter alia*, that neither Petitioner’s expert in this proceeding (Dr. Nielson) nor its expert in the co-pending district court litigation,

Dr. Kevin Butler, disputes that the Appendix to the '639 provisional supports the written description for linking (Sur-reply 7–9 (citing, e.g., Ex. 2015, 42; Ex. 1003; Paper 20, 3–7; Paper 22, 1));⁹ that “Federal Circuit precedent . . . requires only that the specific reference [required under 37 C.F.R. § 1.78(a)(2)] indicate the priority claim with reference to the application numbers of the priority documents, and that the Office therefore correctly found that the priority chain was properly recited in the '648 application and that no correction was necessary (*id.* at 9–10 (citing *Medtronic Corevalve, LLC v. Edwards Lifesciences Corp.*, 741 F.3d 1359, 1365–66 (Fed. Cir. 2014); *Carotek, Inc. v. Kobayashi Ventures, LLC*, 875 F. Supp. 2d 313, 334–35 (S.D.N.Y. 2012); Ex. 1001, 1:7–19; Ex. 2005, 1:16–20; Ex. 2006, 1:6–11)); and that Petitioner’s collateral estoppel arguments are incorrect (*id.* at 10–13 (citing 35 U.S.C. § 315(e); *Papst Licensing GMBH v. Samsung Elecs. Am., Inc.*, 924 F.3d 1243, 1249–50 (Fed. Cir. 2019); *Soverain Software*, 778 F.3d at 1315–16)).

3. Analysis

Upon review of the parties’ respective arguments, we remain unpersuaded that the challenged claims are entitled to a priority date any earlier than the December 22, 1997, filing date of the '648 application.

As an initial matter, as the Board explained in *Polaris Wireless, Inc. v. TruePosition, Inc.*, Case IPR2013-00323, Paper 9 (PTAB Nov. 15, 2013), there is no presumption of earlier priority where the specifications of the earlier applications are not the same. *See id.* at 29. Patent Owner does not

⁹ As discussed in Section II.F. below, the parties address the Appendix and Dr. Butler’s testimony in more detail in the Motion to Terminate and Opposition thereto.

dispute that the specification of the '844 patent differs from the specifications of the Earlier Applications. Thus, "Patent Owner must come forward with evidence and argument . . . showing why the challenged claim is supported by the written description of the priority application." *Nintendo of Am. Inc. v. iLife Techs.*, Case IPR2015-00106, Paper 12 at 16 (PTAB Apr. 29, 2015). Moreover, "[i]t is not sufficient for purposes of the written description requirement of § 112 that the disclosure, when combined with the knowledge in the art, would lead one to speculate as to modifications that the inventor might have envisioned, but failed to disclose." *Lockwood v. Am. Airlines, Inc.*, 107 F.3d 1565, 1572 (Fed. Cir. 1997). For the reasons set forth below, we are not persuaded in this case by Patent Owner's arguments and evidence presented in support of its contention that the challenged claims are sufficiently supported by the written description of the '639 provisional.

In its Response, as cited above, Patent Owner contends that the '639 provisional describes a system that is "connected to the Internet," receives "a Downloadable . . . from [an] external network, analyzes the Downloadable to generate and link a DSP, and then forwards the Downloadable to a web client (or discards it)." PO Resp. 19–20. The '639 provisional does not expressly disclose a "web server," but Patent Owner cites Dr. Jaeger's testimony as "explaining that the external network 105 [depicted in Figures 1 and 2 of the '639 provisional] includes a web server, such as the web server for 'cnn.com,' which is connected to the Internet." *Id.* at 20 (citing Ex. 2016 ¶ 59). Patent Owner then concludes that "[b]ecause a DSP is linked to the Downloadable before the Downloadable reaches a web client, the 'linking . . . before a web server makes a

Downloadable available to a web client’ is disclosed in the ’639 Provisional Application.” *Id.* We disagree with Patent Owner’s conclusion. Assuming that the external network would include a web server, as Dr. Jaeger contends (Ex. 2016 ¶ 59), and further assuming, for example, that code scanner 325 of the ’639 provisional can be said to “link” the Downloadable and corresponding DSP data by virtue of its “send[ing]” both to second comparator 330, as Patent Owner argues (*see, e.g.*, PO Resp. 21), Patent Owner does not identify a *web server* that makes the Downloadable available to web clients *after* such linking, as required by the challenged claims. Petitioner’s arguments above reflect that the “web server” identified by Dr. Jaeger’s testimony cited by Patent Owner is in “external network 105,” whereas the DSP is generated and allegedly linked to the Downloadable only after the Downloadable is received “from the external network”—and by extension, therefore, *after* the web server makes the Downloadable available. *See* PO Resp. 19–20 (citing Ex. 2016 ¶ 59).

Similarly, in the annotated versions of Figures 3 and 6 of the ’639 provisional provided in the Patent Owner Response and reproduced in Section II.C.2. above, Patent Owner has added a box labeled “Web Server” in each figure, *prior to* annotation boxes labeled as “Receiving a Downloadable.” PO Resp. 21 (annotated Figure 3 of the ’639 provisional), 23 (annotated Figure 6 of the ’639 provisional). In describing those annotated figures, Patent Owner cites the ’639 provisional as disclosing that ID generator 315 (which, in the annotated version of Figure 3, Patent Owner identifies as the component that generates and links the DSP) “receives Downloadables *from external computer network 105*” and that “a Downloadable is received from a web server and then the inspector will

generate an ID for the Downloadable.” PO Resp. 21 (emphasis added) (quoting Ex. 2015, 15), 23 (emphasis added) (citing Ex. 2015, 20). Based on Patent Owner’s arguments, we are not persuaded that the ’639 provisional demonstrates possession by the inventors of linking a DSP to the Downloadable “*before a web server makes the Downloadable available to web clients.*” Rather, by the time that the DSP has been linked to the Downloadable (e.g., by code scanner 325, in Patent Owner’s mapping), we understand that the Downloadable has already been received—and hence “made available”—by the web server. Notably, this conclusion does not depend on *where* the “linking” occurs—particularly, whether it may occur at a network gateway, a focus of the parties’ claim construction arguments (*see, e.g.*, Pet. 25–27; PO Resp. 10–12; Section II.B.5, *supra*)—but instead on *when* it occurs relative to the Downloadable being made available to clients. Further, having considered the alleged Appendix to the ’639 provisional (Ex. 2015, 37–45), including portions cited by Dr. Butler in related litigation (*see* Ex. 2047), we are not persuaded that anything in the Appendix demonstrates possession of linking “before a web server makes the Downloadable available to web clients.”

In conclusion, therefore, we are not persuaded that the inventors were in possession of the subject matter of the challenged claims at any time prior to the December 22, 1997, filing date of the ’648 application.

As an independent basis for concluding that the challenged claims are not entitled to any earlier priority date, we also agree with Petitioner (Pet. 14–15) that Patent Owner has not demonstrated compliance with the requirement of 37 C.F.R. § 1.78(a)(2) that “[a]ny nonprovisional application claiming the benefit of one or more prior filed copending nonprovisional

applications . . . must contain or be amended to contain in the first sentence of the specification following the title a reference to each such prior application . . . *indicating the relationship of the applications*” (emphasis added). The “PRIORITY REFERENCE TO RELATED APPLICATIONS” section of the ’844 patent fails to satisfy that requirement, because it does not “indicat[e] the relationship” of the ’639 provisional, ’388 application, and ’097 application to the ’844 patent. Ex. 1001, 1:7–19. We recognize that Patent Owner attempted to correct that defect by filing a Request for Certificate of Correction, but, as noted above, the Office denied that Request in relevant part, stating that the Request was improper “[w]ith respect to the alleged error in Col. 1, 1st para”—referring to the requested amendment to the “PRIORITY REFERENCE TO RELATED APPLICATIONS.” Ex. 2008, 2.

Despite Patent Owner’s assertion that the Office denied the request because “no correction was necessary” (*see* Sur-reply 9), we agree with Petitioner that the Office instead denied the request because it was an improper attempt to perfect a claim of priority that was not found in the original application (Reply 15–16; *see* Ex. 2008, 2 (stating that “comparison of the printed patent with the corresponding location in the application file reveals that there is no discrepancy”)). Further, we do not understand the Federal Circuit’s decision in *Medtronic Corevalve* to stand for the Patent Owner’s proposition that § 1.78(a)(2) “requires only that the specific reference indicate the priority claim with reference to the application numbers of the priority documents.” *See* Sur-reply 9–10. Although that case makes clear that § 1.78(a)(2) “requires precise details in priority claims down to the ‘application number (consisting of the series code and serial

number),” it does not eliminate the additional requirement in that provision to “indicat[e] the relationship of the applications.” *Medtronic Corevalve*, 741 F.3d at 1365–66. Indeed, it expressly refers to that requirement. *Id.* at 1365 (“The relationship between the applications is whether the instant application is a continuation, divisional, or continuation-in-part of the prior nonprovisional application. An example of a proper benefit claim is ‘this application is a continuation of prior Application No. ___, filed ___.’” (quoting MPEP § 201.11 (8th ed rev. 5, Aug. 2006)), 1367 (noting that “Medtronic used language suggested by the M.P.E.P. in a contrary manner, and additionally failed to disclose the correct relationships between the applications at issue”).

D. Ground 1: Obviousness of Claims 1, 15, and 41 over Abadi

1. Overview of Abadi

Abadi, titled “Method and Apparatus for Annotating a Computer Program to Facilitate Subsequent Processing of the Program,” relates “generally to the processing of mobile[] computer programs, and more particularly to annotating such programs to assist downstream processing.” Ex. 1008, 1:8–11.

An embodiment of a networked system disclosed by Abadi is illustrated in Figure 2, reproduced below.

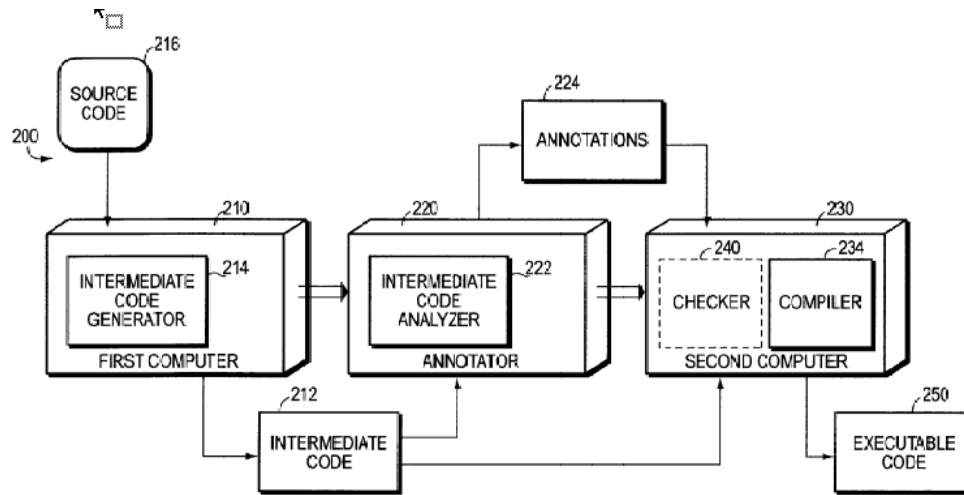


FIG. 2

Figure 2 of Abadi, above, is a block diagram schematic of an embodiment of Abadi's invention. *See* Ex. 1008, 3:1–2.

With reference to Figure 2, Abadi explains that annotator 220 is coupled to first computer 210 and second computer 230. *Id.* at 3:8–10. First computer 210 includes intermediate code generator 214, which converts received source code 216 into intermediate code 212. *Id.* at 3:18–21. According to Abadi, source code 216 can be written in any programming language, such as Java, C or C++, whereas intermediate code 212 is machine-independent, meaning that “code 212 itself does not run on any particular computer without further processing, e.g., interpreting or compiling.” *Id.* at 3:21–29. Annotator 220 includes intermediate code analyzer 222, which analyzes intermediate code 212 and produces annotations 224. *Id.* at 3:36–38. The analysis can include, for example, mapping variables to registers, determining a control flow of code 212, determining methods for optimizing code 212, checking that data structures are initialized and that code 212 is syntactically well-formed, contains valid references to data structures, data fields, and other code, and verifying that

operations performed by code 212 do not underflow or overflow the stack. *Id.* at 3:38–46. From annotator 220, code 212 and annotations 224 pass to second computer 230. *Id.* at 3:47–48. Abadi discloses that although code 212 and annotations 224 are illustrated as being separately forwarded to second computer 230, intermediate code analyzer 222 can annotate code 212 itself so that annotations 224 are placed in code 212, producing an annotated intermediate and resulting in code 212 and annotations 224 arriving concurrently at second computer 230. *Id.* at 3:48–54. Alternatively, the annotations can be stored and transmitted separately. *Id.* at 2:32–36.

Abadi discloses, in particular, that intermediate code analyzer 222 statically analyzes code 212 to produce annotations 224 that may provide “any information about the code that can be obtained from static analysis” and “fall into at least two types: annotations that characterize properties of the code; and annotations that are in the form of a formal proof of the code.” *Id.* at 4:60–5:2. Abadi further discloses that annotations 224 characterizing properties of code 212 provide second computer 230 with information that assists in subsequent processing. *Id.* at 5:6–9. For example, annotations 224 may contain information about how the code accesses memory, allowing second computer 230 to conclude that the code is safe to execute, or can contain information allowing the code to be optimized for increased performance. *Id.* at 5:12–19. Exemplary annotations can characterize the behavior of a single code statement, a block of code statements, or the flow of execution of the entire code 212. *Id.* at 5:27–29.

Additionally, Abadi teaches that “annotations can be generated at a number of locations in a network before being transmitted to the second computer system [i.e., the client].” *Id.* at 2:25–27. For example, Abadi

explains that “the first computer system that produced the code can also produce the annotations.” *Id.* at 2:27–31. Alternatively, “a third computer system between the first and second computer systems, for example, a computer on a firewall protecting the second computer system from receiving potentially harmful programs, can generate and transmit the annotations to the second computer system.” *Id.* at 2:37–42.

In a second embodiment, depicted in Figure 3 reproduced below, Abadi teaches that the annotator can reside on a server.

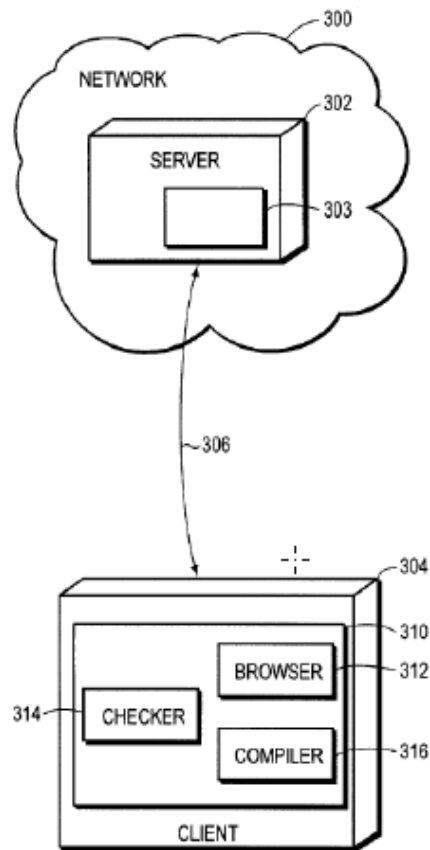


FIG. 3

Figure 3, above, is a block diagram of an exemplary application of Abadi's invention. *Id.* at 3:3–4.

As depicted in Figure 3, server 302 includes annotator 303, which statically analyzes and annotates each applet attached to a web page supported by server 302 before the applet is sent to client 304. *Id.* at 6:46–55. Client 304, in turn, includes memory 310 for storing browser 312 (e.g., “Netscape CommunicatorTM” or “Microsoft Internet ExplorersTM” [sic]), annotation checker 314, and compiler 316. *Id.* at 6:58–64. According to Abadi, if server 302 is trusted by client 304, client 304 can process the annotated intermediate code, including by checking the safety of the applet before executing (i.e., interpreting or compiling) the applet. *Id.* at 7:6–11. For example, Abadi explains, the annotations may provide information from which browser 312 on client 304 can determine whether the applet is safe to execute on client 304 or can suggest register allocations that help browser 312 execute the applet more efficiently. *Id.* at 7:12–17. If client 304 does not trust the applet produced by server 302, client 304 analyzes the annotations along with the applet to make sure that the applet would not perform any unwanted operations when run on client 304. *Id.* at 7:18–22. In that case, browser 312 would reject the applet if checker 314 determines that the annotations are false. *Id.* at 7:22–25.

2. *Analysis*

a. *“receiving by an inspector a Downloadable” / “an inspector system”*

Claims 1 and 41 each recite “receiving by an inspector a Downloadable.” Claim 15 recites “an inspector system.” Petitioner asserts that Abadi teaches these limitations by virtue of its teaching of “an annotator (i.e., ‘an inspector’) that receives ‘executable application programs’ such as

‘applet[s] attached to [a] web page’ (i.e., ‘Downloadable’).” Pet. 32 (citing Ex. 1008, 6:51–53), 52 (citing Ex. 1008, 2:25–42, 6:51–53), 56.

Petitioner contends, for example, that annotator 220 is an “inspector,” as recited in the claims, “because it statically analyzes code to create annotations that ‘characterize the behavior of a single code statement, a block of code statements, or the flow of execution of the entire code 212.’” *Id.* at 33 (citing Ex. 1008, 5:27–29). Moreover, Petitioner contends, “Abadi discloses that ‘annotator 220’ receives ‘code coming from the first computer 210,’” and “Abadi teaches that the ‘code’ received by the annotator can be in various forms,” including “intermediate code.” *Id.* at 33–34 (citing Ex. 1008, 3:10–12, 3:19–38, 8:35–37). Petitioner explains, with support from its declarant, Dr. Nielson, that “[c]omputer programs converted to ‘intermediate code’ are . . . ‘Downloadables’ because they are ‘executable application programs that are downloaded from a source computer and run on the destination computer,’” and “[i]ndeed, after conversion, the intermediate code is the computer program and can be executed.” *Id.* at 34 (citing Ex. 1003 ¶ 72). Petitioner also points out that Abadi additionally “discloses that in some instance the ‘intermediate code’ can be the final executable form, even without compiling it.” *Id.* at 34–35 (citing Ex. 1008, 2:20–22, 4:32–33, 5:12, 6:26, 7:9–11).

Petitioner further contends that Abadi’s “annotator 303,” which “statically analyzes and annotates, according to the principles of the invention, each applet attached to the web page,” also is an “inspector.” *Id.* at 36 (citing Ex. 1008, 6:41–53). Petitioner points out that the ’844 patent itself specifically identifies “applets” as examples of “Downloadables,” as well as that Abadi explains that the applets processed by annotator 303 are

executable application programs. *Id.* at 37 (citing Ex. 1001, 1:49–56; Ex. 1008, 7:4–11; Ex. 1003 ¶ 78).

Patent Owner argues that Abadi’s annotator 220 is not an “inspector” because it does not perform the same functions as the inspector or the content inspection engine recited in the challenged claims. PO Resp. 40. In particular, Patent Owner contends, both Abadi’s “intermediate code” and its “source code” are non-executable, and accordingly, annotator 220 does not receive a “Downloadable” under the agreed-upon construction of that term. *Id.* Patent Owner further contends that, even if the Board were to find that the intermediate code or source code of Abadi is executable code, Petitioner’s argument that Abadi teaches this claim element still fails because the agreed-upon construction of Downloadable requires that the Downloadable be downloaded from a source computer and run on a destination computer. *Id.* at 41–42. In contrast, according to Patent Owner, “the entire purpose of the Abadi invention is to make sure the original intermediate code is *never* meant to run on a destination computer because intermediate code is instrumented with annotations, thus creating a brand new intermediate code.” *Id.* at 42 (citing Ex. 2016 ¶ 91). Thus, Patent Owner contends, “the original intermediate code that is downloaded from a source computer can never run on the destination computer.” *Id.*

With respect specifically to what Petitioner refers to as Abadi’s “network embodiment,” Patent Owner contends, Abadi “uses the annotations to modify the intermediate code using a compiler to create a brand new executable code such that the executable code on the destination computer is not what was downloaded from the source computer.” *Id.* (citing Ex. 1008, 4:44–50). Specifically, Patent Owner argues, “the

annotations will be used in the compilation to change the intermediate code,” and therefore, “the original intermediate code does not meet the definition of the ‘Downloadable’.” *Id.* at 42–43. Indeed, Patent Owner contends, “Petitioner never identifies a destination computer that the original intermediate code is meant to run on because Abadi modifies the original intermediate code, and [thus] it can never be the same Downloadable that was provided by a ‘source computer.’” *Id.* at 43 (citing Ex. 2016 ¶ 91).

Similarly, with respect to what Petitioner terms Abadi’s “server embodiment,” Patent Owner contends that “Petitioner admits that the annotated intermediate code (i.e., the annotated applet) sent to the client 304 is not the same as the intermediate code (i.e., original applet).” *Id.* (citing Pet. 35–37). Thus, “[t]he original applet is never sent to the client, as [the] annotator modifies the applet to instrument the code using annotations,” and “the original intermediate code cannot be a ‘Downloadable’ because it is never meant to be run on a destination computer by Abadi’s design.” *Id.*

In its Reply, Petitioner responds that Patent Owner focuses on only a specific embodiment of Abadi and “ignores that Abadi explicitly teaches that it ‘is to be understood that the practice of the principles of the invention is *not limited to intermediate code*, but rather that annotations can be generated for various other types of code, such as, for example, source code, machine code, machine-dependent or machine-independent code, high-level or low-level code, assemble code, etc.’” Reply 17–18 (quoting Ex. 1008, 3:28–35). Thus, Petitioner contends, the “argument that ‘intermediate code’ is not executable and does not meet the agreed construction of ‘Downloadable’ is not dispositive.” *Id.* at 18. Further, Petitioner contends, “Patent Owner’s argument that Abadi’s ‘machine code’ is not executable is

belied by both its own expert's admission that 'machine code is often the same thing as binary code' and can be executable ([Ex. 1037,] 154:11–18), and the teachings of Abadi itself," which "confirms that 'code 212' . . . is executable when it is machine code." *Id.* at 18–19 (citing Ex. 1008, 5:13–16). Still further, Petitioner contends,

Even if Abadi were limited to "intermediate code"—which it is not—Patent Owner's arguments are still unpersuasive because, contrary to Patent Owner's contentions, the "intermediate code" described in Abadi is "executable" within the meaning of the '844 Patent, notwithstanding that Abadi's intermediate code must be interpreted or compiled. As discussed above, the '844 Patent identifies Java Applets as an example of a "Downloadable["](EX-1001 at, *e.g.*, claim 5), and Java Applets are code that must be interpreted by the Java Virtual Machine or JIT-compiled to be executed at the destination. EX-1037 at 152:16–153:5; EX-1044 at 2. The '844 Patent similarly identifies JavaScript and Visual Basic scripts as example Downloadables (EX-1001 at, *e.g.*, claims 7–8), which are both also code that must be interpreted to be executed. EX-1037 at 116:14–117:4; EX-1039. Thus, the '844 Patent expressly contemplates that "Downloadables" may be in intermediate code form that must be interpreted or compiled to be executed. This is also consistent with Abadi, which teaches that "'executing' means interpreting or compiling." EX-1008 at 7:10.

Reply 19. Lastly, Petitioner disputes Patent Owner's argument that annotating the 'intermediate code' modifies it such that the file downloaded from the source computer is not the same as the file run on the destination computer, pointing out that Abadi discloses embodiments in which annotations are stored and transmitted separately from the code. *Id.* at 19–20 (citing Ex. 1008, 8:34–36). And in any event, Petitioner asserts, "even if Abadi only described embodiment[s] where annotations are attached to or

embedded in the code, the agreed construction of Downloadable does not preclude these types of modifications.” *Id.* at 20.

In its Sur-reply, Patent Owner responds that Petitioner “misunderstands or mischaracterizes” Abadi, repeating that Abadi does not disclose receiving a Downloadable because the executable code 250 generated at the end of Abadi’s process—whether in the form of intermediate code or machine code—is different from the code that was received. Sur-reply 15–18.

Having considered the parties’ arguments and cited evidence, we are persuaded that Abadi teaches “receiving by an inspector a Downloadable” under the parties’ agreed construction of the term “Downloadable.” *See supra* Section II.B.1. In particular, we are persuaded that Abadi teaches that applets attached to a web page and received and annotated by annotator 303 of server 302 are “executable application program[s]” that may be “downloaded from a source computer” and “run on [a] destination computer.” Ex. 1008, 6:45–7:29. As Petitioner points out, the ’844 patent itself discloses Java applets as “[e]xamples of Downloadables.” Pet. 37 (citing Ex. 1001, 1:49–56); Ex. 1001, 1:62–65. Moreover, as Petitioner also points out, Abadi discloses that “the practice of the principles of the invention is not limited to intermediate code”—which Patent Owner contends is not “executable” and, therefore, cannot constitute a Downloadable—“but rather that annotations can be generated for various other types of code, such as, for example, . . . machine code.” *See* Reply 17–18 (citing Ex. 1008, 3:28–35); PO Resp. 40.

As we explain below, we agree with Patent Owner that Abadi teaches that its code is subsequently modified by embedding annotations within it

(*see, e.g.*, Sur-reply 15), such that the executable code 250 generated at the end of Abadi's process is different from the code that was received, and that, accordingly, Abadi does not teach or suggest that "the Downloadable" is made available to web clients. *See infra* Section II.D.2.c. Nonetheless, we disagree with Patent Owner's argument that the received code "is not, therefore, a Downloadable." *See* Sur-reply 15. Contrary to Patent Owner's argument, the fact that Abadi modifies the code (*e.g.*, "to improve its compilation and performance") does not *necessarily* mean that Abadi's code, at least in certain embodiments,¹⁰ was not executable in the first place.

b. *"generat[ing] [by the inspector] a first Downloadable security profile that identifies suspicious code in [the received/a] Downloadable"*

Claim 1 recites "generating by the inspector a first Downloadable security profile that identifies suspicious code in the received Downloadable." Claims 15 and 41 similarly recite "generat[ing] a first Downloadable security profile that identifies suspicious code in a Downloadable" and "generating a first Downloadable security profile that identifies suspicious code in the received Downloadable," respectively. Petitioner asserts that Abadi teaches or suggests these limitations by virtue of its teaching that "the annotator (*i.e.*, 'the inspector') generates annotations (*i.e.*, 'a first Downloadable security profile') that 'characterize the behavior of a single code statement,' including, *e.g.*, mis-typed assignments, unauthorized operations, or code that overflows or underflow the stack (*i.e.*, 'that identifies suspicious code in the received Downloadable')."

¹⁰ To be sure, we acknowledge that not all received code disclosed by Abadi is "executable." *See, e.g.*, Ex. 1008, 3:24–28 ("[C]ode 212 itself does not run on any particular computer without further processing.").

Pet. 37–38, 54–56. Petitioner contends, for example, that “Abadi teaches a system to ‘facilitate *security checks* and code analyses,’ and for ‘*improved accuracy of the security checks*.’” *Id.* at 38 (citing Ex. 1008, 1:56–58).

Petitioner further contends, “the ‘annotations’ disclosed by Abadi ‘identif[y] suspicious code’ because they identify code that is hostile, potentially hostile, undesirable, or potentially undesirable” and “are clearly intended to alert users to code that is potentially hostile or undesirable (i.e., ‘suspicious’).” *Id.* at 40 (citing Ex. 1008, 1:56–58, 2:22–25, 5:6–11).

Petitioner explains, with support from its declarant, Dr. Nielson, that the examples Abadi provides of annotations flagging mis-typed assignments, unauthorized operations, and code that overflows or underflow the stack reflect techniques that are commonly employed by hackers to evade security restrictions or to shift control to malicious code, and would thus, given the general knowledge of a person of ordinary skill in the art, identify suspicious code. *Id.* at 40–42 (citing Ex. 1003 ¶¶ 85–88; Ex. 1008, 3:12–16, 3:31–32, 3:36–46, 5:27–29, 5:36–53).

Still further, Petitioner contends, “Abadi’s annotator ‘deems’ certain code as suspicious.” *Id.* at 43. In particular, Petitioner argues:

Abadi specifically discloses that annotations are intended to be used for security purposes, and the disclosed system will indicate the presence of code that has been deemed suspicious, such as code performing *unauthorized* operations. [Ex.] 1008, 2:21–23. A POSITA would understand that, in order to make a determination as to whether a particular operation is “unauthorized,” the system must have “deemed” certain operations authorized and others unauthorized. [Ex.] 1003 ¶ 90.

As another example, Abadi discloses that the annotator may create “annotations that *characterize* properties of the code”—as opposed to annotations that may simply list properties of the code. [Ex.] 1008, 5:4; *see also id.* [at] 5:27–29

(annotations “can *characterize the behavior* of a single code statement, a block of code statements, or the flow of execution of the entire code 212”). To “characterize” the code, the annotator must be able to make determinations about the code, including, e.g., deeming it suspicious. [Ex.] 1003 ¶ 91. Moreover, Abadi specifically performs analysis to determine whether the code is harmful or potentially harmful—for example by “verifying that operations performed by the code 212 do not underflow or overflow the stack.” [Ex.] 1008, 3:44–46. Thus, while the ultimate decision of whether executable code is safe or will be blocked is left to the client (just like in the ’844 Patent), Abadi’s annotator deems code as suspicious and the annotations accordingly identify such suspicious code to facilitate the client’s decision. [Ex.] 1003 ¶ 91.

Id. at 43–44.

In its Response, Patent Owner argues that Abadi does not disclose or render obvious generating a “Downloadable security profile” because Abadi’s annotations do not “identify suspicious code.” PO Resp. 43–47. Patent Owner argues, “[t]he purpose of Abadi is to improve performance of a program, not to identify suspicious code,” and Abadi’s “annotations 224 are not [a] Downloadable security profile.” *Id.* at 43–44. Indeed, Patent Owner contends, the design of Abadi is such that it would optimize a malicious program to run better, as an objective of Abadi “is to enhance program code . . . with supplementary information that will help subsequent processing stages.” *Id.* (quoting Ex. 1008, 1:63–65) (citing Ex. 2016 ¶¶ 87, 100). Moreover, Patent Owner contends, “Abadi specifically notes that its annotation process is separate and apart from elements of the system that protect against ‘harmful programs.’” *Id.* at 44–45 (citing Ex. 1008, 4:35–43). And although a person of ordinary skill in the art would understand that the annotation process could potentially be implemented on the same

computer that protects a second computer from harmful programs, the annotation process itself does not provide such protection. *Id.* at 45 (citing Ex. 2016 ¶ 101). Further, although Abadi refers to “unauthorized operations,” a person of ordinary skill in the art “would understand that in the context of Abadi, the ‘unauthorized operations’ would not be suspicious code, but a way to improve efficiency and reduce errors for a program by rewriting the program so that it does not result in an error by calling for items not available to it.” *Id.* at 46 (citing Ex. 1008, 4:24–34 (describing “unauthorized operations” in the context of whether the intermediate code is “well-formed”); Ex. 2016 ¶ 98). Similarly with respect to Abadi’s disclosure that annotations can identify code that overflows or underflows the stack, Patent Owner contends that such “identification . . . is to make sure the program operates correctly, not that it would detect an attack on a user’s computer.” *Id.* at 46–47 (citing Ex. 2016 ¶ 99). In fact, Patent Owner argues, “if a malware had an overflow or underflow stack problem, the overflow or underflow stack problem would be corrected, thus allowing the malware to operate as intended.” *Id.* at 47 (citing Ex. 2016 ¶ 99). Finally, Patent Owner contends that Petitioner misunderstands the purpose of Abadi, which is designed to improve performance by fixing program code using annotations, not about preventing malware. *Id.* (citing Pet. 42–44). Even where Abadi mentions “safety,” Patent Owner argues, it does so only “in the context of type-checking, run-time optimization, control flow graphs, mapping data structures, method offsets, and proof carrying code”—i.e., in the context of improving performance, not detection of malware. *Id.* (citing Ex. 1008, 5:6–6:22; Ex. 2016 ¶ 74, 103).

In its Reply, Petitioner responds that Patent Owner’s arguments that Abadi’s annotations do not identify suspicious code are unpersuasive for at least two reasons: First, Petitioner points out, “suspicious” code is stated in the ’844 patent to include not only “potentially hostile or hostile” code but also “undesirable, or potentially undesirable code” (Reply 21 (quoting Ex. 1001, 5:21–23)); and second, “Abadi identifies *both* inadvertent bugs *and* intentionally malicious code” (*id.* at 22). In support of the latter argument, Petitioner argues that Abadi incorporates by reference a paper by Necula (Ex. 1033) that, in turn, states that “even trusted persons . . . can make errors occasionally or even act maliciously” (*id.* at 2), and therefore, Petitioner contends, “[t]he purpose of Abadi’s annotations . . . is therefore not limited to identifying bugs/errors and expressly includes protecting against malicious code as well” (Reply 22). Petitioner further contends “Abadi’s ‘security checks’ improve upon prior art security checks, which ‘may leave open doors *for attack*’” and that “[t]his is why Abadi teaches the creation of annotations that can be used to, among other things, ‘indicate whether running the code would perform *unauthorized* operations.’” *Id.* at 22–23 (quoting Ex. 1008, 1:49–50, 2:22–24) (citing Ex. 1008, claim 14). According to Petitioner, “[t]here can be no reasonable dispute that Abadi’s ‘security checks[]’ . . . constitutes [sic] identifying suspicious code because unauthorized operations are necessarily undesirable and therefore ‘suspicious’ as defined by the ’844 Patent.” *Id.* at 23.

Having considered the parties’ arguments and cited evidence, we are not persuaded by a preponderance of the evidence that Abadi teaches or suggests generating a “Downloadable security profile” that “identifies

suspicious code” in a Downloadable, as required by each of the challenged claims.

As explained in the overview in Section II.D.1. above, Abadi’s annotators 220 and 303 analyze code and produce annotations that include, for example, mapping variables to registers, determining a control flow of code, determining methods for optimizing code, checking that data structures are initialized and that the code is syntactically well-formed, contains valid references to data structures, data fields, and other code, and verifying that operations performed by code 212 do not underflow or overflow the stack. Ex. 1008, 3:38–46. However, we find no indication in Abadi that the annotators identify anything as being “suspicious”—i.e., hostile, potentially hostile, undesirable, or potentially undesirable. Rather, the annotations produced by annotators 220 and 303 are passed along with the analyzed code to second computer 230 or client 304, respectively, which then uses the information in the annotations to process the code. *Id.* at 2:8–14, 2:19–25, 3:47–48, 7:3–22.

To the extent that Abadi’s references to “security checks,” “safe[ty],” “undesirable operations,” and “unwanted operations” suggest “suspicious” operations, as Petitioner contends, we are not persuaded that such operations are “identified” as being suspicious in the annotations themselves. *See, e.g.*, Ex. 1008, 1:23–25, 1:56–60, 2:21–24, 5:12–16, 7:9–10, 7:12–14. Rather, it is second computer 230 or client 304 that is described by Abadi in each instance as making such determinations after receiving the annotations. *Id.* Indeed, Abadi expressly provides that “the user of the second computer 230 does not trust the first computer 210 or any code 212 coming from the second computer,” specifically because “the user of the second computer

230 does not know whether an executable form of the code 212 will perform any unauthorized operations.” *Id.* at 3:11–16. Abadi likewise discloses that “annotations 224 produced by the first computer 210 are untrusted” and that second computer 230 accordingly needs to verify the integrity of the annotations. *Id.* at 4:13–17. With regard to the disclosed server embodiment, Abadi similarly discloses that “[t]ypically . . . client 304 does not trust the applet produced by the server 302,” which applet comprises “annotated intermediate code” annotated by annotator 303 of server 302, and accordingly “would analyze the annotations along with the applet to make sure that the applet would not perform any unwanted operations when the applet runs.” *Id.* at 7:18–22, 6:51–57, 7:1–5. Even where “server 302 is trusted by client 304,” Abadi discloses that client 304 itself “check[s] the safety of the applet before executing (i.e., interpreting or compiling) the applet.” *Id.* at 7:6–11.

In other words, Abadi’s teachings consistently suggest that any identification of unsafe or undesirable operations is not made by the annotations that Petitioner maps to the claimed Downloadable security profile, but instead such identification only occurs once the annotations are processed by the second computer system or client, which Petitioner maps to the claimed “web client.” *See, e.g.,* Pet. 37, 45, 49. Read in this context, we do not understand statements such as those in Abadi’s Abstract, cited by Petitioner, that “[o]ther exemplary annotations can indicate that running the executable form of the code would perform an unauthorized operation at the second computer system” and that “[t]he second computer system could then reject the code instead of performing subsequent processing on the code” (Ex. 1008, Abstract (cited at Pet. 28)) to mean that the annotations include

an identification of suspicious code, but only that the annotations provide information from which the second computer could determine whether to reject the code or to perform subsequent processing. Petitioner does not address this deficiency of Abadi or cite any persuasive evidence that a person of ordinary skill in the art would have understood Abadi to teach that the annotations themselves identify suspicious code.

c. *“linking [by the inspector] the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients”*

Claim 1 recites “linking by the inspector the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients receiving by an inspector a Downloadable.” Claims 15 and 41 each similarly recite “linking the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients receiving by an inspector a Downloadable.” Petitioner asserts that Abadi teaches or suggests these limitations, for example, by virtue of its teaching that the “the annotator (i.e., ‘the inspector’) associates (i.e., ‘links’) the annotation (i.e., ‘the first Downloadable security profile’) with the Downloadable before the first computer or the server makes the Downloadable available to other computers in the network” or “before the applet (i.e., ‘Downloadable’) is sent to the client.” Pet. 44–51 (citing, e.g., Ex. 1008, 1:60–67, 2:27–41, 3:48–65, 4:8–16, 6:45–46, 6:51–57, 6:65–7:9; Ex. 1003 ¶¶ 94, 95, 97–99, 101, 103, 105), 55 (relying for claim 15 on the same arguments and evidence as for claim 1), 56 (same for claim 41). Petitioner contends, for example, that “Abadi teaches that the annotator can be included on a web server,” and that “Abadi expressly discloses that the annotations are generated ‘before the

applet is sent to the client.”” *Id.* at 49–50 (emphasis omitted) (citing Ex. 1008, 6:51–57, 6:54–55). Petitioner further contends, “Abadi teaches that the annotations may be ‘linked’ to the Downloadable in a variety of ways—including by ‘add[ing] the annotations to the code,’ ‘plac[ing] the annotations] in the code,’ or ‘stor[ing] the annotations separately from the code and transmit[ting] the annotations and code separately.”” *Id.* at 44–45 (citing Ex. 1008, 2:31–37, 3:48–54). Relying on Dr. Nielson’s testimony, Petitioner further contends “a POSITA would recognize that there are a finite number of ways to implement Abadi’s annotator on the ‘server,’” the more efficient of which being “to generate the annotations before the Downloadable is made available on the server.” *Id.* at 50 (citing Ex. 1003 ¶ 103). Petitioner also points out that, “[n]otably, Abadi’s architecture of having the annotator reside on a server mirrors the disclosures of the ’844 Patent,” and “[i]ndeed, the ’844 Patent specifically discloses that the inspector and web server may be part of an integral site.” *Id.* at 51 (citing Ex. 1001, 3:47–52).

In response, Patent Owner argues that Abadi does not create a Downloadable until after the Abadi system uses the annotations to compile executable machine code 250. PO Resp. 48. Moreover, Patent Owner contends, even if it is determined that Abadi’s intermediate code or source code is a “Downloadable,” “Abadi still does not teach this claim element because the ‘Downloadable’ identified by Petitioner is never made available to web clients.” *Id.* (citing Ex. 2016 ¶¶ 105–107). Rather, in each instance identified by Petitioner, Abadi modifies the intermediate code to optimize performance, and the original intermediate code is never the one that is made available to a web client. *Id.* at 48–49 (citing Pet. 44–51; Ex. 1008, 4:44–

51). According to Patent Owner, “[b]y ‘adding’ code to modify its ultimate compilation (or interpretation), the final executable program can never be the same as the original because the whole purpose of Abadi is to optimize performance by making the Downloadable to have less errors at compilation.” *Id.* at 49. Finally, Patent Owner contends, “[t]o the extent that Petitioner is arguing that the web client is the second computer 230 is the web client and that it can receive intermediate code 212 separately from annotations 224, that argument fails” because “the intermediate code 212 is transformed into executable code by compiler 234 using modifications from annotations 224 to guide its compilation/interpretation into executable code 250, thus intermediate code 212 is never made available to second computer.” *Id.* (citing Ex. 1008, 4:44–51).

Petitioner responds in its Reply that “Patent Owner does not dispute that Abadi ‘links’ the annotations . . . that Abadi generates for the code that is being analyzed (*i.e.*], the ‘Downloadable’), nor does Patent Owner dispute that this linking occurs before the code is made available to the web client by Abadi’s web server under either parties’ construction of this limitation.” Reply 23 (citing PO Resp. 48–49). Further, Petitioner contends, although Patent Owner “argues that the Downloadable is not made available to web clients at all because the annotations modify the original Downloadable,” “at least one embodiment of Abadi indisputably does not modify the code” and, “in any event, the agreed construction of Downloadable does not preclude modification in transit by attaching a security profile.” *Id.* at 23–24 (citing PO Resp. 48–49; Ex. 1008, 8:34–36, Fig. 2).

In its Sur-reply, Patent Owner responds that “Petitioner is unable to identify a single instance of where the alleged Downloadable in Abadi is not modified.” Sur-reply 20. Indeed, Patent Owner contends, “in Petitioner’s Reply, Petitioner identifies Abadi at 8:34–36 as an example of where Abadi does not modify the alleged Downloadable,” but “as shown in the same claim, the same alleged ‘computer program’ received from the first computer is modified by ‘transforming and optimizing the code into an executable code’ using the annotations.” *Id.* (citing Reply 23; Ex. 1008, claim 15). And accordingly, Patent Owner argues, “the original Downloadable is never made available to the web client using Abadi’s system.” *Id.*

Having considered the parties’ arguments and cited evidence, we are not persuaded by a preponderance of the evidence that Abadi teaches or suggests linking “the Downloadable security profile” to “the Downloadable” before a web server makes the Downloadable available to web clients, as required by each of the challenged claims.

As an initial matter, the antecedent basis for the phrase “the Downloadable security profile” is the “Downloadable security profile that identifies suspicious code” in a Downloadable, and, as explained in Section II.D.2.b. above, we are not persuaded that Abadi teaches or suggests a Downloadable security profile “that identifies suspicious code” at all. Even apart from that determination, however, we also are not persuaded that Abadi teaches or suggests linking annotations to the Downloadable before a web server makes “the Downloadable” available to web clients.

In particular, as mentioned in Section II.B.1. above, we agree with Patent Owner that the ordinary meaning of the claim language requires that

the Downloadable that is to be made available to web clients in the “linking” steps of the challenged claims must be the same executable application program that is downloaded from the source computer within the agreed construction. That this must be the case is supported by the express language of the challenged claims, in which “the Downloadable” that is “ma[de] . . . available to web clients” in the final step of each of independent claims 1 and 41 has its antecedent basis in the first claim step, “receiving . . . a Downloadable.” Accordingly, both steps are understood to refer to the same Downloadable—i.e., the same “executable application program.” Similarly, in the context of independent claim 15, the antecedent for “the Downloadable” that is made available to web clients is the Downloadable in which suspicious code is identified by the content inspection engine. We also agree with Patent Owner that the claimed “linking” of a DSP “to the Downloadable” does not compel a different understanding, as we are not persuaded that linking requires modification of the code of the Downloadable itself. As explained in Section II.D.2.a. above, we are persuaded that Abadi teaches “receiving . . . a Downloadable” insofar as it discloses applets attached to a web page and received and annotated by annotator 303 of server 302. Ex. 1008, 6:45–7:29. In connection with that disclosure, however, Abadi specifically discloses that the “annotated intermediate code representing the applet” is sent by server 302 to client 304 and that “client 304 can process the annotated intermediate code according to the annotations *embedded in the code*.” *Id.* at 7:3–9 (emphasis added). Because Abadi describes the annotations as “embedded” in the code, and not simply linked as required by the challenged claims, we agree with Patent Owner that the “annotated intermediate code representing the applet” is not

the same Downloadable that is received by Abadi's annotator, but is instead a modified version thereof. *See* PO Resp. 48–49.

In contrast with the embodiment shown in Figure 3, we recognize that Abadi also describes the embodiment depicted in “FIG. 1”¹¹ as showing annotations 224 being “forwarded to the second computer 230” “separately” from code 212. Ex. 1008, 3:48–52; *see also id.* at 2:32–36 (stating that “the first computer system may add the annotations to the code and send both together to the second computer system, *or store the annotations separately from the code and transmit the annotations and code separately*” (emphasis added)). Abadi further explains, however, that, although the figure shows the annotations “to be separately forwarded . . . , the intermediate code analyzer 222 can annotate the code 212 so that the annotations 224 are placed in the code 212, producing an annotated intermediate code.” *Id.* at 3:48–54. According to Abadi, “[p]lacing the annotations 224 in the code 212 displaces the need for locally caching the analysis,” thereby addressing a shortcoming of prior art systems in which “only the computer with the cached analysis benefited from th[e] analysis.” *Id.* at 3:55–65. As mentioned above, Abadi describes the server embodiment shown in Figure 3 only as including annotations “embedded in the code,” and Petitioner does not cite any persuasive evidence that a person of ordinary skill in the art would have understood Abadi to teach or suggest any differently, particularly in light of Abadi's disclosure referenced above of the purported

¹¹ Based on the reference numerals 212, 222, and 224 recited in the cited passage, as well as on the content of the described subject matter, we understand this description to relate to Figure 2 of Abadi rather than to Figure 1.

advantages of placing annotations “in the code” and thereby modifying the code.

d. Conclusion Regarding Ground 1

For the foregoing reasons, Petitioner has not shown by a preponderance of the evidence that any of claims 1, 15, and 41 is unpatentable over Abadi.

E. Ground 2: Obviousness of Claim 15 over Abadi and Gryaznov

1. Overview of Gryaznov

Gryaznov, titled “Scanners of the Year 2000: Heuristics,” is an article describing the use of “heuristic analysis” for virus scanning, discussing approaches to heuristical program analysis, and comparing various scanners employing heuristics for both virus detection and false alarms rate.

Ex. 1009, 2.

2. Analysis

Petitioner contends that, to the extent that the Board finds that Abadi does not inherently disclose or alone render obvious “memory storing a first rule set” as recited in claim 15, that limitation would have been obvious over the combination of Abadi and Gryaznov. Pet. 56–60. Petitioner does not allege that Gryaznov teaches the “Downloadable” limitations recited in claim 15 so as to cure the deficiency of Abadi identified in the above discussion of Ground 1, but instead relies on Abadi alone and incorporates by reference its discussion from Ground 1 for those limitations. Pet. 60–61. Accordingly, we conclude for the same reasons set forth in the analysis of Ground 1 that Petitioner has not established by a preponderance of the evidence that claim 15 is unpatentable over the combined teachings of Abadi and Gryaznov.

F. Patent Owner's Motion to Terminate

Following Petitioner's service of an expert report in related district court litigation, Patent Owner requested authorization to file a motion to terminate this proceeding or for supplemental briefing, alleging that Petitioner's expert in that litigation, Dr. Butler, had taken contradictory positions vis-à-vis positions taken by Petitioner and Dr. Nielson in this proceeding with regard to the priority date of the '844 patent. We held a conference call, a transcript of which is filed as Exhibit 1030, and issued an Order (Paper 19) authorizing supplemental briefing on this issue, in response to which Patent Owner filed a Motion to Terminate (Paper 20) and Petitioner filed an Opposition (Paper 22).

In the Motion to Terminate, Patent Owner alleged that Dr. Butler had in his expert report (portions of which are filed in this proceeding as Exhibits 1031 and 2047) mapped the teachings in a particular document to claim 41 of the '844 patent as evidence that the claimed invention was on sale more than one year before the '844 patent's filing date. Paper 20, 1–2. According to Patent Owner, the document relied upon by Dr. Butler was not an offer for sale, but was instead a confidential invention disclosure that was faxed to Patent Owner's patent counsel and filed with the '639 provisional application. *Id.* at 2. A copy of the document is filed as Exhibit 2048 in this proceeding and at pages 37–39 of Exhibit 2015 as an “Appendix” to the '639 provisional. *Id.* Patent Owner contends, “[a]s a result, Dr. Butler actually analyzed part of the '639 Provisional that was filed and he concluded that it contained the written description support for the challenged claims,” thus confirming that the challenged claims are entitled to an earlier priority date. *Id.* In its Opposition, Petitioner responds, *inter alia*, that

Patent Owner has omitted the “Priority Date” section of Dr. Butler’s report, in which Dr. Butler expressly opined that the ’639 provisional did not provide sufficient written description support for the ’844 patent; that Dr. Butler’s statements relied on by Patent Owner expressly state that the mapping of the fax to claim 41 would apply only if the court were to adopt Patent Owner’s overbroad construction of the claim’s “linking” limitation; and that, regardless of Dr. Butler’s opinions, Patent Owner failed to properly claim priority to the ’639 provisional. Paper 22, 1–10.

For the reasons provided in Section II.C. above, we are not persuaded that the challenged claims are entitled to a priority date any earlier than the December 22, 1997, filing date of the ’648 application for the ’822 patent, notwithstanding the arguments made in Patent Owner’s Motion to Terminate and evidence cited in it. Moreover, in view of our conclusion that Petitioner has not established by a preponderance of the evidence that the challenged claims are unpatentable, we dismiss Patent Owner’s Motion to Terminate as moot.

G. Motions to Exclude Evidence

In *inter partes* review proceedings, documents are admitted into evidence subject to an opposing party asserting objections to the evidence and moving to exclude the evidence. 37 C.F.R. § 42.64. Petitioner and Patent Owner each move to exclude certain exhibits to which they previously objected. Pet. Mot. Excl.; PO Mot. Excl. In particular, Petitioner moves to exclude Exhibits 2020, 2021, 2025, 2033, 2037, 2038, 2041, and 2046, all of which relate to alleged objective evidence of nonobviousness, i.e., secondary considerations (Pet. Mot. Exclude 2–10); and Patent Owner moves to exclude Exhibits 1003, 1004, 1010–1012, 1015–1025, 1027, 1028,

and 1031–1044 (PO Mot. Excl. 1–15). We dismiss as moot both of the parties’ motions because (1) even without excluding Exhibits 1003, 1010, 1037, 1040, and 1041, we conclude that Petitioner has failed to establish by a preponderance of the evidence that any of the challenged claims are unpatentable; and (2) we do not refer to or rely upon any of Exhibits 1004, 1011, 1012, 1015–1025, 1027, 1028, 1031–1036, 1038, 1039, 1042–1044, 2020, 2021, 2025, 2033, 2037, 2038, 2041, and 2046 in reaching our conclusions herein.

H. Petitioner’s Motion to File Under Seal and Enter Proposed Protective Order

Petitioner moves under 37 C.F.R. §§ 42.14 and 42.54 to file certain documents and to enter the Office’s default Protective Order. Mot. to Seal 2–3; *see* Office Patent Trial Practice Guide, App’x B, 77 Fed. Reg. 48,756, 48,769–71 (Aug. 14, 2012). In particular, Petitioner requests to file Exhibits 1036 and 1038, as well as certain portions of its Reply that quote or directly refer to Exhibits 1036 and 1038, under seal, based on Patent Owner’s designation of those Exhibits as confidential. *Id.*¹² Petitioner represents that it has met and conferred with Patent Order regarding entry of the Default Protective Order pursuant to 37 C.F.R. § 42.54(a) and that the parties agree thereto. *Id.* at 2.

There is a strong public policy in favor of making information filed in an *inter partes* review open to the public, especially because the proceeding determines the patentability of claims in an issued patent and, therefore, affects the rights of the public. *See Garmin Int’l, Inc. v. Cuozzo Speed*

¹² As noted in footnote 1, *supra*, Petitioner filed a redacted version of its Reply as Paper 25 in this proceeding.

Techs., LLC, Case IPR2012-00001 (PTAB Mar. 14, 2013) (Paper 34). Under 35 U.S.C. § 316(a)(1) and 37 C.F.R. § 42.14, the default rule is that all papers filed in an *inter partes* review are open and available for access by the public; a party, however, may file a concurrent motion to seal and the information at issue is sealed pending the outcome of the motion. It is, however, only “confidential information” that is protected from disclosure. 35 U.S.C. § 316(a)(7); *see* Office Patent Trial Practice Guide, 77 Fed. Reg. at 48,760. The standard for granting a motion to seal is “for good cause.” 37 C.F.R. § 42.54(a). The party moving to seal bears the burden of proof in showing entitlement to the requested relief, and must explain why the information sought to be sealed constitutes confidential information. 37 C.F.R. § 42.20(c).

In reviewing Exhibits 1036 and 1038, we conclude that they may contain confidential information. Accordingly, we are persuaded that good cause exists to have those Exhibits and the identified portions of the Reply remain under seal, and the Motion to File Under Seal and Enter Proposed Protective Order is granted.

The Office Patent Trial Practice Guide provides:

Expungement of Confidential Information: Confidential information that is subject to a protective order ordinarily would become public 45 days after final judgment in a trial. There is an expectation that information will be made public where the existence of the information is . . . identified in a final written decision following a trial. A party seeking to maintain the confidentiality of information, however, may file a motion to expunge the information from the record prior to the information becoming public. § 42.56. The rule balances the needs of the parties to submit confidential information with the public interest in maintaining a complete and understandable file history for public notice purposes. The rule encourages

parties to redact sensitive information, where possible, rather than seeking to seal entire documents.

77 Fed. Reg. at 48,761.

Consequently, 45 days from entry of this decision, all information subject to a protective order will be made public by default. In the interim, Patent Owner may file a motion to expunge any such information that is not relied upon in this Decision. *See* 37 C.F.R. § 42.56.

III. CONCLUSION

Petitioner has not shown by a preponderance of the evidence that claims 1, 15, and 41 of the '844 patent are unpatentable.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that claims 1, 15, and 41 of the '844 patent have not been shown to be unpatentable;

FURTHER ORDERED that Petitioner's Motion to Terminate is dismissed;

FURTHER ORDERED that Petitioner's Motion to Exclude Evidence is dismissed;

FURTHER ORDERED that Patent Owner's Motion to Exclude Evidence is dismissed;

FURTHER ORDERED that Petitioner's Motion to File Under Seal and Enter Proposed Protective Order is granted; and

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

In summary:

Claims	35 U.S.C. §	Reference(s)	Claims Shown Unpatentable	Claims Not Shown Unpatentable
1, 15, 41	103	Abadi		1, 15, 41
15	103	Abadi, Gryaznov		15
Overall Outcome				1, 15, 41

IPR2019-00026
Patent 6,154,844

For PETITIONER:

Michael R. Fleming
Jonathan S. Kagan
Rebecca L. Carson
IRELL & MANELLA LLP
mfleming@irell.com
jkagan@irell.com
rcarson@irell.com

For PATENT OWNER:

James Hannah
Jeffrey Price
Michael Lee
KRAMER LEVIN NAFTALIS & FRANKEL LLP
jhannah@kramerlevin.com
jprice@kramerlevin.com
mhlee@kramerlevin.com
svdocketing@kramerlevin.com