

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SOPHOS LTD. and SOPHOS INC.,
Petitioners

v.

OPEN TEXT INC.,
Patent Owner

Case No. PGR2023-00031
Patent No. 11,409,869

PATENT OWNER'S PRELIMINARY RESPONSE

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	THE BOARD SHOULD EXERCISE ITS DISCRETION TO DENY INSTITUTION UNDER 35 U.S.C. § 314(A).....	3
A.	<i>Fintiv</i> Factor 1: The District Court Has Not Granted and Will Not Grant a Stay.....	6
B.	<i>Fintiv</i> Factor 2: The District Court Trial is Expected Well Before the Projected Statutory Deadline.....	11
C.	<i>Fintiv</i> Factor 3: The Parties Have Invested Significant Resources in The Parallel District Court Proceeding.....	14
D.	<i>Fintiv</i> Factor 4: The Issues and Parties Overlap Significantly.....	17
E.	<i>Fintiv</i> Factor 5: Both Parties Are Parties in the Parallel Litigation.	18
F.	<i>Fintiv</i> Factor 6: The Petition Does Not Present “Compelling Evidence of Unpatentability.”	20
III.	THE BOARD SHOULD DENY INSTITUTION BECAUSE PETITIONER IS NOT LIKELY TO SUCCEED.	21
A.	Background	23
1.	The State of the Art.....	23
2.	The ’869 Patent	25
3.	The Board’s Rejection of Sophos’s Earlier Petition Against the Parent ’869 Patent.....	30
4.	Claim Construction	31
5.	The Level of Ordinary Skill	33

B.	Ground 1: Ideses Fails to Render Any of the Challenged Claims Obvious	33
1.	Overview of Ideses.....	34
2.	Ideses Fails to Render Any of the Independent Claims Obvious Because It Does Not Teach or Suggest Turning Features On or Off at the Classification Stage Based on the Executable Being Classified	37
3.	Ideses Fails to Render Any of the Dependent Claims Obvious	41
C.	Ground 2: Mori Fails to Render Any of the Challenged Claims Obvious.....	41
1.	Overview of Mori.....	42
2.	Mori Fails to Render Any of the Independent Claims Obvious Because It Does Not Teach or Suggest Turning Features On or Off at the Classification Stage Based on the Executable Being Classified	44
3.	Mori Fails to Render Any of the Dependent Claims Obvious	50
D.	Ground 3: The Combination of Ideses and Mori Does Not Render Claims 1-59 Obvious	50
E.	Ground 4: The Phrase “The Static Data Points” Does Not Render Claims 1-59 Indefinite	51
F.	Ground 5: The “String” Limitations Do Not Make Dependent Claims 8, 14, 17, 29, 43, or 57 Indefinite.....	55
G.	Ground 6: The “String” Limitations in Dependent Claims 8, 14, 17, 29, 43, and 57 Have Adequate Written Description	58
H.	There are objective indicia of non-obviousness.....	60
IV.	CONCLUSION.....	61

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Altera Corp. v. PACT XPP Techs.</i> , AG, No. 14-CV-02868-JD, 2015 WL 4999952 (N.D. Cal. Aug. 21, 2015).....	52
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 11 (PTAB Jun. 1, 2021)	<i>passim</i>
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 15 (PTAB May 13, 2020)	11
<i>Apple v. Seven Networks</i> , IPR2020-00255, Paper 13 (PTAB July 28, 2020)	14
<i>Ariad Pharms., Inc. v. Eli Lilly & Co.</i> , 598 F.3d 1336, 1351 (Fed. Cir. 2010)	59, 60
<i>Blast Motion, Inc. v. Zepp Labs, Inc.</i> , No. 15-CV-700 JLS (NLS), 2017 WL 476428 (S.D. Cal. Feb. 6, 2017).....	52
<i>Cizion, LLC v. Kerr Machine Co.</i> , PGR2020-00065, Paper 10 (PTAB Dec. 3, 2020).....	13
<i>Commscope Techs. v. Dali Wireless, Inc.</i> , IPR 2022-01242, Paper 23(PTAB Feb. 27, 2023).....	21
<i>Eclinical Works, LLC v. Decapolis</i> LLC, IPR2022-00229, Paper 10 (PTAB Apr. 13, 2022).....	12
<i>In re Glob. IP Holdings LLC</i> , 927 F.3d 1373 (Fed. Cir. 2019)	59
<i>Google LLC v. EcoFactor, Inc.</i> , IPR2021-00454, Paper 9 (PTAB July 22, 2021).....	18

Patent Owner’s Preliminary Response
to Petition for *Post-Grant Review*
Patent No. 11,409,869

<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966).....	60
<i>In re Huang</i> , 100 F.3d 135 (Fed. Cir. 1996)	60
<i>InfoGation Corp. v. ZTE Corp.</i> , No. 16-CV-01901-H-JLB, 2017 WL 1821402 (S.D. Cal. May 5, 2017).....	52, 53
<i>Intell. Ventures II LLC v. BITCO Gen. Ins. Corp.</i> , No. 615CV59LEADCASE, 2016 WL 125594 (E.D. Tex. Jan. 11, 2016).....	53
<i>Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA (W.D. Tex. Dec. 22, 2022).....	7
<i>Leo Pharm. Prods., Ltd. v. Rea</i> , 726 F.3d 1346 (Fed. Cir. 2013)	60
<i>Luxshare Precision Industry v. Amphenol Corp.</i> , IPR2022-00132, Paper 10 (PTAB May 3, 2022)	10
<i>Magellan Midstream Partners L.P v. Sunoco Partners Mktg. & Terms. L.P.</i> , IPR2019-01445, Paper 12 (PTAB Jan. 22, 2020)	12
<i>mCom IP, LLC v. Cisco Sys., Inc.</i> , No. 6:22-00261-ADA, slip op. (W.D. Tex. Oct. 20, 2022).....	7
<i>Motorola Mobility LLC v. Ironworks Patents, LLC</i> , IPR2021-00421, Paper 11 (PTAB July 22, 2021).....	11
<i>Nautilus Inc. v. Biosig Instruments, Inc.</i> , 572 U.S. 898 (2014).....	52, 57
<i>NHK Spring Co., Ltd. v. Intri-Plex Techs., Inc.</i> , IPR2018-00752, Paper 8 (PTAB Sept. 12, 2018) (<i>precedential</i>).....	12

Patent Owner's Preliminary Response
to Petition for *Post-Grant Review*
Patent No. 11,409,869

<i>Nintendo Co. v. Gamevice, Inc.</i> , IPR2020-01197, Paper 13 (PTAB Jan. 12, 2021)	10
<i>Nokia of Am. Corp. v. Godo Kaisha IP Bridge</i> , IPR2022-00755, Paper 9 (PTAB Nov. 2, 2022).....	19
<i>NXP USA, Inc. v. Impinj, Inc.</i> , PGR2022-00005, Paper 20 (PTAB Aug. 25, 2022)	13
<i>Regeneron Pharms. Inc. v. Novartis Pharma AG</i> , IPR2020-01317, Paper 15 (PTAB Jan. 15, 2021)	15, 16
<i>Robert Bosch Tool Corp. v. SD3, LLC</i> , IPR2016-01751 (PTAB Mar. 22, 2017)	60
<i>S3 Inc. v. NVIDIA Corp.</i> , 259 F.3d 1364 (Fed. Cir. 2001)	57
<i>Samsung Elecs. Co. v. California Institute of Tech.</i> , IPR2023-00131, Paper 10 (PTAB May 4, 2023)	12
<i>Samsung Elecs. Co. v. Clear Imaging Research LLC</i> , IPR2020-01399, Paper 13 (PTAB Feb. 3, 2021).....	10, 11
<i>Sand Revolution II, LLC v. Cont'l Intermodal Group-Trucking LLC</i> , IPR2019-01393, Paper 24 (PTAB June 16, 2020)	16, 17
<i>Sonrai Memory Ltd. v. Kingston Tech. Co., Inc.</i> , No. 6:21-cv-1284-ADA (W.D. Tex. Oct. 18, 2022)	7
<i>Sophos Ltd. v. Webroot Inc.</i> , IPR2023-00528, Paper 1, 17-64 (PTAB Feb. 7, 2023)	2, 30
<i>Sophos Ltd. v. Webroot Inc.</i> , IPR2023-00528, Paper 14 (PTAB Aug. 14, 2023).....	<i>passim</i>
<i>Sotera Wireless, Inc. v Masimo Corp.</i> , IPR2020-01019, Paper 12 (PTAB Dec. 1, 2020)	18

Patent Owner’s Preliminary Response
to Petition for *Post-Grant Review*
Patent No. 11,409,869

Tuna Processors, Inc. v. Hawaii Int’l Seafood, Inc.,
327 F. App’x 204 (Fed. Cir. 2009)53

Twinstrand Biosciences, Inc. v. Guardant Health, Inc.,
IPR2022-01115, Paper 14 (PTAB Dec. 30, 2022)41, 50

Vizio, Inc. v. Maxell, Ltd.,
IPR2022-01459, Paper 7 (PTAB Apr. 11, 2023)19

Statutes

35 U.S.C. § 314(a)2, 3, 20

37 C.F.R. § 42.100(b)30

AMERICA INVENTS ACT, Congress. H. Rept.
112-98 Part 1 (June 1, 2011).....1

Other References

Director’s Memo, *Interim Procedure for Discretionary Denials in AIA*
Post-Grant Proceedings with Parallel District Court Litigation
(June 21, 2022).....*passim*

EXHIBIT LIST

Exhibit	Description
2001	Declaration Of Sam Malek, Ph.D.
2002	<i>Curriculum Vitae</i> of Sam Malek, Ph.D.
2003	<i>Webroot, Inc., Open Text Inc., v. AO Kaspersky Lab et al.</i> , Case No. 6:22-cv-00243-ADA-DTG, Claim Construction Order, Dkt. 391 (W.D. Tex July 25, 2023)
2004	<i>Sonrai Memory Ltd. v. Kingston Tech. Co. and Kingston Tech. Corp.</i> , Case No. 6:21-cv-1284-ADA, Order Denying Kingston Tech. Co. and Kingston Tech. Corp.’s Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 94 (W.D. Tex., Oct. 18, 2022)
2005	<i>mCom IP, LLC v. Cisco Systems, Inc.</i> , Case No. 6:22-cv-00261-ADA, Order Denying Defendant’s Motion to Stay Pending <i>Inter Partes</i> Review, Dkt. 42 (W.D. Tex., Oct. 20, 2022)
2006	<i>Intellectual Ventures I LLC et al. v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA, Order Denying Hewlett Packard Enterprise Company’s Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 104 (W.D. Tex., Dec. 22, 2022)
2007	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Dec. 9, 2022 Transcript of Motions Hearing, Dkt. 133 (W.D. Tex. Dec. 15, 2022) (excerpted)
2008	Dani Kass, “Catching Up On Patent Litigation With Judge Albright” (Law360 Mar. 14, 2023); https://www.law360.com/pulse/articles/1582438/print?section=pulse/courts
2009	U.S. District Courts—Federal Court Management Statistics—Profiles—During the 12-Month Periods Ending March 31, 2018 Through 2023, “Table N/A—U.S. District Courts—Combined Civil

Patent Owner's Preliminary Response
to Petition for *Post-Grant Review*
Patent No. 11,409,869

Exhibit	Description
	and Criminal Federal Court Management Statistics (March 31, 2023)", available https://www.uscourts.gov/sites/default/files/data_tables/fcms_na_distprofile0331.2023.pdf
2010	<i>Webroot, Inc., Open Text Inc., v. AO Kaspersky Lab et al.</i> , Case No. 6:22-cv-00243-ADA-DTG, Order Granting Fifth Amended Scheduling Order, Dkt. 252 (W.D. Tex., Mar. 29, 2023)
2011	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-00240-ADA, Second Amended Complaint for Patent Infringement, Dkt. 76 (W.D. Tex., Oct. 24, 2022)
2012	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-00240-ADA-DTG, Defendant Sophos Ltd.'s Preliminary Invalidity Contentions For Patents Plaintiffs Added by Amendment (W.D. Tex., Feb. 21, 2023) (excerpted)
2013	<i>Webroot, Inc. and Open Text Inc., v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 6:22-cv-00241-ADA-DTG, CrowdStrike's Preliminary Invalidity Contentions (Patents Asserted by Amendment) (W.D. Tex., Feb. 21, 2023) (excerpted)
2014	<i>Webroot, Inc. and Open Text Inc., v. Trend Micro Inc.</i> , Case No. 6:22-cv-00239-ADA-DTG, Defendant's Preliminary Invalidity Contentions (W.D. Tex., Feb. 21, 2023) (excerpted)
2015	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab</i> , Case No. 6:22-cv-00243-ADA-DTG, Defendant's AO Kaspersky Lab's Preliminary Invalidity Contentions (W.D. Tex., Feb. 21, 2023) (excerpted)
2016	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab</i> , Case No. 6:22-cv-00243-ADA-DTG, Opening Claim Construction Brief From: [Defendants], Dkt. 86 (W.D. Tex., Oct. 28, 2022)
2017	<i>Microsoft Computer Dictionary</i> , p. 36 (Fifth Ed. Microsoft 2012)

Patent Owner's Preliminary Response
to Petition for *Post-Grant Review*
Patent No. 11,409,869

Exhibit	Description
2018	<i>Decimal - Binary - Octal - Hex – ASCII Conversion Chart</i> , available at https://www.eecis.udel.edu/~amer/CISC651/ASCII-Conversion-Chart.pdf .
2019	Microsoft, <i>String to numeric value functions</i> , available at https://learn.microsoft.com/en-us/cpp/c-runtime-library/string-to-numeric-value-functions?view=msvc-140

Open Text Inc. ("Patent Owner" or "Open Text") submits this Preliminary Response to the Petition of Sophos Ltd. and Sophos Inc. (collectively "Petitioner" or "Sophos") seeking post-grant review of U.S. Patent No. 11,409,869.

I. INTRODUCTION

The Board should deny institution for two reasons: (1) discretionary denial under *Fintiv* is appropriate because the '869 patent is part of an exceptionally complex parallel litigation that is scheduled for trial months before a final written decision on this Petition would be due, and (2) the petition is not "more likely than not" to succeed because it merely parrots arguments the Board already rejected when it denied Sophos's earlier petition for IPR of the '869 patent's parent (U.S. Patent No. 10,599,844), as well as arguments already rejected in parallel litigation.

First, the Board should deny institution under *Fintiv* because instituting this review would undermine Congress's stated goals in creating these proceedings: to provide "quick and cost effective alternatives to litigations" and "limit unnecessary and counterproductive litigation costs." H.R. Rep. No. 112-98, pt. 1, at 40 (2011), 2011 U.S.C.C.A.N. 67, 69. Petitioner and other defendants have raised the same prior art and arguments in parallel litigation before the Western District of Texas ("WDTX"). Trial on those arguments is scheduled to occur nearly four months *before* a final written decision on this petition would be due. The district court has

stated that it will not stay cases due to parallel post-grant reviews when, as here, the court can resolve the same issues earlier than the Board. Consequently, a post-grant review will not provide an “alternative” to litigation, let alone a “quick and cost effective” one. Instead, instituting the PGR would generate the unnecessary and counterproductive litigation costs that Congress sought to avoid.

Second, the Board should deny institution under 35 U.S.C. §§ 324(a) and 325(d) because the petition merely parrots prior art arguments that the Board already rejected, when refusing to institute Sophos's earlier IPR petition on the '869 patent's parent. The PGR grounds (1-3) here press the same prior art and arguments Petitioner pressed in its failed IPR petition against the '844 Patent. (*Compare* Petition 15-86 (arguing that Ideses and Mori render obvious all claims of the '869 patent) *with Sophos Ltd. v. Webroot Inc.*, IPR2023-00528, Paper 1, 17-64 (PTAB Feb. 7, 2023) (arguing that Ideses and Mori render obvious all claims of the '844 patent).) The Board refused to institute IPR against the '844 patent because neither Ideses nor Mori teach or suggest turning features on or off *at the classification stage*, as required by the '844 patent's claims. *Sophos Ltd.*, IPR2023-00528, Paper 14, 18 (PTAB Aug. 14, 2023) (“Sophos IPR Denial”). As shown below, the claims of the child '869 patent require that same feature. Consequently, Ideses and Mori cannot render the claims of the '869 patent obvious either.

Like the prior art grounds (1-3), the Section 112 grounds (4-6) largely parrot arguments that have already been rejected—this time by the district court. (*See* EX-1008 at 52-54 (arguing that “the static data points” is indefinite); EX-2003 at 19 (rejecting argument that “the static data points” is indefinite); EX-2016 at 59-65 (arguing that the “string” limitations are indefinite); EX-1007 18 (rejecting argument that the “string” limitations are indefinite).) The Board should likewise reject these arguments.

II. THE BOARD SHOULD EXERCISE ITS DISCRETION TO DENY INSTITUTION UNDER 35 U.S.C. § 314(A).

The '869 patent is one of dozens of patents asserted in claims and counterclaims among six different parties in a series of cases pending in the Western District of Texas. The district court has already issued its claim constructions and is on track to adjudicate the same prior art and invalidity grounds raised in the petition months before the Board would issue a final written decision on this Petition. Institution of a parallel PGR proceeding trailing so far behind the district court would needlessly increase the cost of resolving the underlying dispute. Under these circumstances, the Board should exercise its discretionary authority to deny institution under 35 U.S.C. § 324(a).

Because Congress's goal in enacting post-grant proceedings was “to ... limit unnecessary and counterproductive litigation costs,” Director Vidal has

instructed the Board not to institute post grant reviews that would “increase, rather than limit, litigation costs.” Director’s Memo, Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation (“Guidance”) at 1 (June 21, 2022).

Instituting this PGR would cause just such an increase. The reasons are easy to see: a district court stay is out of the question, and the district court will consider the very same arguments presented in the petition. Accordingly, the Board and the district court will labor to decide the very same issues with the district court well ahead of—and scheduled to reach a final decision months before—the Board. The year-long parallel tracks of two adjudicatory bodies will “increase, rather than limit, litigation costs,” create “potential conflict between the ... (PTAB) and district court proceedings,” and generally increase “the potential for inefficiency and gamesmanship.” Guidance at 1. This is the situation that Director Vidal sought to avoid when she instructed the Board to take “a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review.” *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 6 (PTAB Mar. 20, 2020).

Every one of the *Fintiv* factors weighs for discretionary denial:

(1) A stay of the co-pending district court litigation—involving 39 patents and five defendants—is highly unlikely. Petitioner cannot avoid that trial because it

has petitioned for review of only some of the patents asserted against it. This PGR will not narrow the issues on which discovery and trial must proceed because the '869 patent is asserted against four other defendants, not all of whom have petitioned for review of the '869 patent. Moreover, none of Petitioner's codefendants have agreed not to litigate the validity of the '869 patent in the district court if the Board institutes this Petition.

(2) Trial in the district court is expected almost four months *before* a final written decision. The district court case remains on track. The claims of the '869 patent have been construed, and discovery is well underway.

(3) The parties and district court have already invested heavily in carrying the parallel litigation past claim construction and well into discovery.

(4) The issues raised in the petition are the same as those the district court has before it, and Petitioner's narrow stipulation will not change that because the stipulation does not apply to all of Petitioner's co-defendants, who have raised the same prior art before the district court.

(5) Petitioner will get its chance to make its arguments before the district court because Petitioner is a defendant in the parallel litigation.

(6) As shown below, the petition's grounds of unpatentability are weak.

Thus, every *Fintiv* factor here weighs for discretionary denial. The district court is well on its way to an efficient resolution of the disputes between Patent Owner, Petitioner, and Petitioner's codefendants. Both the goals of the AIA and the general welfare of the patent system would be best served by denying institution.

A. *Fintiv* Factor 1: The District Court Has Not Granted and Will Not Grant a Stay.

The first *Fintiv* factor asks “whether [1.] the court granted a stay or [2.] evidence exists that one may be granted if a proceeding is instituted.” *Fintiv*, IPR2020-00019, Paper 11 at 6.¹ Here, the undisputed answer to both questions is *no*: no stay has been requested or ordered, and Petitioner has identified no evidence

¹ The decision in *Fintiv* addresses various scenarios that may indicate whether a stay may be granted. *Fintiv*, IPR2020-00019, Paper 11 at 6-9. However, nothing in *Fintiv* suggests that these are the only scenarios that a panel may consider. *Fintiv* does not hold that an actual denial of a stay is the only consideration relevant to this factor. Rather, *Fintiv* contemplates that any type evidence may be relevant. For example, *Fintiv* holds that one way to “support the arguments that ... a stay may be less likely, and instituting would lead to duplicative costs” is to show the “work completed by the parties and court in the parallel proceeding.” *Fintiv* IPR2020-00019, Paper 11 at 10.

that one will (or even might) be. To the contrary, all evidence indicates that a stay is unlikely.

First, “the most important factor” the court considers in its stay determinations is issue simplification²—and here, a stay would not simplify anything. The validity of the ’869 patent will be litigated in the consolidated action, whether this petition is instituted or not, because the patent is asserted against four

² See, e.g., EX-2004, *Sonrai Memory Ltd. v. Kingston Tech. Co., Inc.*, No. 6:21-cv-1284-ADA, slip op. at *4-5 (W.D. Tex. Oct. 18, 2022) (denying a stay following IPR institution because issue simplification was the “most important factor” and the Court had already held a *Markman* hearing and “expended significant resources”); EX-2005, *mCom IP, LLC v. Cisco Sys., Inc.*, No. 6:22-00261-ADA, slip op. at *5 (W.D. Tex. Oct. 20, 2022) (finding “simplification is not likely” where the IPR did not challenge all of the claims asserted in the district court, since the other claims would still require a trial); EX-2006, *Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.*, No. 6:21-cv-00596-ADA, slip op. at *7 (W.D. Tex. Dec. 22, 2022)(denying a stay following IPR institution , as even IPR estoppel would not simplify the invalidity case, given the defendants’ reliance on prior art systems).

other defendants, none of whom have promised not to pursue invalidity in the district court based on this institution decision.

Moreover, the validity of the '869 patent will be litigated in the district court alongside issues regarding 38 *other* patents asserted by and against Petitioner and *four other* defendants—issues that the district court is highly unlikely to delay deciding in favor of letting the Board proceed with *this* trial. The case against Petitioner alone involves twenty patents—thirteen asserted by Patent Owner and seven asserted in counterclaims by Petitioner.

The table below illustrates the patents, parties, and petitions involved in this highly complex litigation:

**Table 1. Patent Owner's Asserted Patents
Against Each WDTX Defendant and Resulting Review Petitions**

Patent	Kaspersky	CrowdStrike	Forcepoint	Sophos	Trend Micro
11,409,869	IPR2023-01160	IPR2023-01053	Asserted; no IPR	PGR2023-00031	Asserted; no IPR
8,201,243	Asserted; no IPR	IPR2023-01052	Asserted; no IPR	IPR2023-00731	Asserted; no IPR
8,719,932	Asserted; no IPR	IPR2023-01051	Asserted; no IPR	IPR2023-00732	Asserted; no IPR
8,418,250	IPR2023-01334	IPR2023-00289	N/A	IPR2023-0699	IPR2023-00655
8,726,389	Asserted; no IPR	IPR2023-00556	Asserted; no IPR	IPR2023-00633	Asserted; no IPR
10,599,844	Asserted; no IPR	Asserted; no IPR	Asserted; no IPR	IPR2023-00528	IPR2023-00662

Patent Owner's Preliminary Response
to Petition for *Post-Grant Review*
Patent No. 11,409,869

Patent	Kaspersky	CrowdStrike	Forcepoint	Sophos	Trend Micro
9,413,721	N/A	N/A	Asserted; no IPR	IPR2023- 00491	N/A
9,578,045	IPR2023- 01199	IPR2023- 00124	N/A	Asserted; no IPR	IPR2023- 00677
10,257,224	IPR2023- 01011	IPR2023- 00126	N/A	Asserted; no IPR	IPR2023- 00657
10,284,591	IPR2023- 00895	IPR2022- 01522	N/A	Asserted; no IPR	IPR2023- 00692
8,856,505	Asserted; no IPR	IPR2023- 01159	N/A	Asserted; no IPR	Asserted; no IPR
8,181,244	N/A	IPR2023- 01158	N/A	Asserted; no IPR	Asserted; no IPR
8,763,123	N/A	IPR2023- 1050	N/A	Asserted; no IPR	N/A
8,438,386	N/A	N/A	IPR2023- 00784	N/A	N/A
10,025,928	N/A	N/A	IPR2023- 00786	N/A	N/A

As U.S. Magistrate Judge Gilliland noted during a December 9, 2022 hearing, this is “a phenomenal case that I’m hoping one day maybe people will write songs about because it is about as complicated as you’re going to get with patent cases.” (EX-2007 52.)

Second, the district court has been clear in public comments that it will not stay cases where, as here, the court can resolve the case as quickly as the Board can:

[T]here’s no unfairness to either side by not staying a case for the PTAB when I can try the case as quickly as they can resolve it.... And so far, I think that’s proven to be true.

(EX-2008 3.) Indeed, as commentators have observed, “Judge Albright almost never stays proceedings to let the PTAB make its decision first.” (*Id.* 2.)

Judge Albright's public comments apply squarely to the situation here. The median time-to-trial in WDTX is 28.2 months. (EX-2009 37.) On that timeline, trial is expected in this case in mid-July 2024. Trial is currently scheduled for August 2024. (EX-2010 8.) The deadline for a final written decision here would not be until many months later (in December 2024). That Judge Albright has made public comments indicating that his practice is to deny a stay in precisely the circumstances presented here is further evidence that he would not stay the WDTX case in response to institution.

Even where no stay has been requested, the Board has found Factor 1 to favor discretionary denial where (as here) “Petitioner present[ed] no evidence that it has requested a stay [] or that a stay is likely.” *Luxshare Precision Industry v. Amphenol Corp.*, IPR2022-00132, Paper 10, at 9 (PTAB May 3, 2022). And such a decision is particularly appropriate where the circumstances show that “a stay ... is unlikely.” *Nintendo Co. v. Gamevice, Inc.*, IPR2020-01197, Paper 13, at 11-12 (PTAB Jan. 12, 2021); *Samsung Elecs. Co. v. Clear Imaging Research LLC*, IPR2020-01399, Paper 13 at 11-12 (PTAB Feb. 3, 2021) (holding Factor 1 may weigh for denial where there is evidence that the district court is unwilling “to avoid duplicative efforts and await

the PTAB's final resolution"); *Motorola Mobility LLC v. Ironworks Patents, LLC*, IPR2021-00421, Paper 11 at 7-8 (PTAB July 22, 2021).

In sum, the district court has not granted a stay and there is no evidence suggesting that the district court would grant a stay of a multi-patent, multi-defendant case so that the Board can reach a decision months *after* the current trial date on *just one* patent. Factor 1 weighs in favor of discretionary denial. *Samsung*, IPR2020-01399, Paper 13 at 11-12; *Motorola*, IPR 2021-00421, Paper 11 at 7-8.

B. *Fintiv* Factor 2: The District Court Trial is Expected Well Before the Projected Statutory Deadline.

Factor 2—the implications of the district court and post grant review schedules—also weighs for denial. “[A] parallel proceeding in an advanced state implicates considerations of efficiency and fairness, which can serve as an independent reason to apply discretion to deny institution.” *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 15 at 11-12, 17 (PTAB May 13, 2020) (informative); Guidance at 1.

Here, a PGR would extend months past the parallel district court litigation. WDTX's median time-to-trial is 28.2 months (EX-2009 37), indicating a trial by mid-July 2024—a date consistent with the currently scheduled trial date of August 19, 2024. (EX-2010 8.) Should the Board institute, its final written decision deadline would be December 6, 2024, nearly four months after the scheduled trial date.

In other words, if the Board institutes here, the proceedings would not produce a result before the trial, and the parties would spend a year expending precisely the “duplicative,” “unnecessary[,] and counterproductive litigation costs” that the AIA was designed to avoid. *Fintiv*, IPR2020-00019, Paper 11 at 5 (citing *Magellan Midstream Partners L.P. v. Sunoco Partners Mktg. & Terms. L.P.*, IPR2019-01445, Paper 12 at 10 (PTAB Jan. 22, 2020)); *see id.*, Paper 15 at 17 (discretionarily denying institution where district court trial was scheduled two months before the final decision deadline); *NHK Spring Co., Ltd. v. Intrix-Plex Techs., Inc.*, IPR2018-00752, Paper 8 at 20 (PTAB Sept. 12, 2018) (*precedential*) (finding that institution of IPR when the district court trial would resolve validity before the PTAB “would not be consistent with an objective of the AIA...to provide an effective and efficient alternative to district court litigation”); *Samsung Elecs. Co. v. California Institute of Tech.*, IPR2023-00131, Paper 10 at 15, 24 (PTAB May 4, 2023) (discretionarily denying institution where “[r]egardless of whether we consider the scheduled trial date or the trial date expected based on time-to-trial statistics, the trial in the underlying litigation would commence several months before the expected date of our final written decision”); *Eclinical Works, LLC v. Decapolis LLC*, IPR2022-00229, Paper 10 at 9 (PTAB Apr. 13, 2022) (discretionarily denying institution

where WDTX trial was scheduled “roughly one to two months before any final decision would have been due”).

Petitioner speculates that the district court may ultimately postpone the scheduled trial date (Petition 95-97), but such speculation carries little weight where Petitioner provides no supporting evidence. Guidance at 8-9; *NXP USA, Inc. v. Impinj, Inc.*, PGR2022-00005, Paper 20 at 8 (PTAB Aug. 25, 2022). Petitioner heavily relies on the Board's decision in *Cizion, LLC v. Kerr Machine Co.*, PGR2020-00065, Paper 10 (PTAB Dec. 3, 2020). (Petition 96-97.) However, *Cizion* was decided before Director Vidal issued her guidance in June 2022. Moreover, *Cizion* particularly credited petitioner's supporting evidence regarding other then-scheduled trials that might conflict with the trial date and now-outdated statistics regarding the Western District of Texas. *Cizion*, PGR2020-00065, Paper 10 at 26. Petitioner presents no such evidence here. In contrast, Patent Owner has presented evidence of a median time-to-trial that is consistent with the scheduled trial date, along with Judge Albright's commitment to quickly trying patent cases.

Because the district court will resolve validity before the Board reaches its final written decision, the second *Fintiv* factor also favors discretionary denial.

C. *Fintiv* Factor 3: The Parties Have Invested Significant Resources in The Parallel District Court Proceeding.

Factor 3—the resources invested in the parallel district court proceeding—favors denial “if, at the time of the institution decision, the district court has issued substantive orders related to the patent.” *Fintiv*, IPR2020-00019, Paper 11 at 9-10; *see also id.* at n.16 (collecting cases). *Fintiv* holds specifically that “claim construction orders may indicate that the court and parties have invested sufficient time in the parallel proceeding to favor denial.” *Id.* at 10.

The parties have invested significant resources in the parallel litigation over the last eighteen months. For example, the parties long ago exchanged infringement and invalidity contentions (in October 2022 and February 2023 respectively), and fact discovery is underway. The Court issued its claim construction order for the '869 Patent on July 25, 2023. (EX-2003.)

The investment in this case is colossal. The district court has already held multiple *Markman* hearings and completed claim construction. Its *Markman* order that included the '869 patent was 19 pages, without even considering the Court's substantive analysis or the additional *Markman* orders issued in this case. (EX-2003.) The Court considered and construed two claim terms of the '869 patent. (*Id.* 19.) The parties have engaged in extensive discovery on the parties' contentions, during which Petitioner has produced more than 3.5 million pages of documents and

Patent Owner has made approximately 122GB of source code available. The Court has held multiple discovery hearings. Patent Owner has noticed more than 35 depositions, and the Board's institution decision is not expected until December 6, 2023, just a month before fact discovery is set to close in the district court proceeding. (EX-2010 6.)

Furthermore, Petitioner's and Patent Owner's experts have already invested an extensive number of hours preparing for their reports by reviewing the patent, prior art, and the defendants' invalidity contentions, as well as traveling to review source code for more than three months. *See Regeneron Pharms. Inc. v. Novartis Pharma AG*, IPR2020-01317, Paper 15 at 17 (PTAB Jan. 15, 2021) (denying institution where the parties had expended considerable resources in claim construction, substantial fact discovery, and preparation for expert reports).

The minimal resources Petitioner invested in its petition overlap with the very same investment in Petitioner's district court invalidity grounds and, therefore, would not "be lost" as Petitioner claims. (Petition 97-98.) Furthermore, institution will do nothing to reduce the substantial expenses in the related litigation, as the district court trial will be held nearly four months before the Board reaches its final written decision.

This case is unlike the *Sand Revolution* decision on which Petitioner relies. (Petition 98.) In *Sand Revolution*, the Markman order was two pages, the case was transferred to a new judge, and the parties had not yet invested in developing the merits. *Sand Revolution II, LLC v. Cont'l Intermodal Group-Trucking LLC*, IPR2019-01393, IPR2019-01393, Paper 24 at 11 (PTAB June 16, 2020). And even then, the Court found that this factor weighed at least marginally in favor of exercising discretion. *Id.* Here, where invalidity contentions were exchanged more than six months ago and the parties and their experts have been reviewing prior art and source code for more than three months, this factor weighs in favor of discretionary denial. *Regeneron*, IPR2020-01317, Paper 15 at 17.

Furthermore, Petitioner's delay in filing this petition further weighs for denial. Petitioner waited more than six months after receiving Patent Owner's infringement contentions before filing its petition. During this time, Petitioner knew that the parties were investing in a rapidly proceeding litigation. Petitioner offers no explanation for its delay, which weighs for denial. *Fintiv*, IPR2020-00019, Paper 11 at 11-12; *Regeneron*, IPR2020-01317, Paper 15 at 17 (finding that even where the petition was filed "less than a month after learning of the complaints," the investment by the parties in the parallel proceeding "outweighs the effort expended so far in this proceeding").

D. *Fintiv* Factor 4: The Issues and Parties Overlap Significantly.

Factor 4—the degree of overlap between the issues in the petition and those before the district court—also supports denial.

Regarding this fourth factor, *Fintiv* explains that “concerns of inefficiency and the possibility of conflicting decisions [are] particularly strong” where “the Board [is] presented with substantially identical prior art arguments ... at issue in district court.” *Fintiv*, IPR2020-00019, Paper 11 at 12. “Accordingly, if the petition includes the same or substantially the same claims, grounds, arguments, and evidence as presented in the parallel proceeding, this fact has favored denial.” *Id.*

Petitioners provides only a limited *Sand Revolution*-style stipulation, stating it will not rely on the *same grounds* in the district court. (Petition 98-99.) Such a stipulation can weigh only “*marginally* in favor of not exercising discretion to deny institution.” *Sand Revolution*, IPR2019-01393, Paper 24 at 11-12.³

Under the circumstances here, Petitioner’s stipulation carries *no weight*. Petitioner and Petitioner’s co-defendants are collectively relying in the litigation on the same prior art references and invalidity grounds asserted in the petition—but

³ All emphases added unless otherwise noted.

offer no stipulation.⁴ Thus, Petitioner's stipulation does not bar the *other defendants* in the parallel litigation from litigating—and forcing Patent Owner to litigate—the very same invalidity challenges before the district court. Not only does this multiply the burden on Patent Owner, it creates the possibility of inconsistent decisions between the district court and the Board.

Because Petitioner's stipulation does not “mitigate[] any concerns of duplicative efforts between the district court and the Board,” or “of potentially conflicting decisions,” Factor 4 weighs in favor of discretionary denial. *Sotera*, IPR2020-01019, Paper 12 at 19; *Google LLC v. EcoFactor, Inc.*, IPR2021-00454, Paper 9 at 12-14 (PTAB July 22, 2021).

E. *Fintiv* Factor 5: Both Parties Are Parties in the Parallel Litigation.

Factor 5—the overlap of the parties between the petition and the parallel litigation—suggests denial because Petitioner is a defendant in the parallel litigation

⁴ Petitioner and Co-defendant CrowdStrike each assert Ideses, Mori, indefiniteness, and lack of written description. (EX-2012 12, 59, 140; EX-2013 37, 40, 64-65.) Co-defendant Trend Micro asserts Ideses, indefiniteness, and lack of written description. (EX-2014 43, 75-77, 89-90.) Co-defendant Kaspersky asserts Ideses. (EX-2015 1-2.)

and, as such, will have a full, fair opportunity to litigate its invalidity contentions. Under *Fintiv*, “[i]f a petitioner is unrelated to a defendant in an earlier court proceeding, the Board has weighed this fact against exercising discretion to deny institution.” *Fintiv*, IPR2020-00019, Paper 11 at 13-14. That is not the case here because Petitioner is a defendant in the litigation. Accordingly, the circumstances raise no fairness concerns that denial would deprive Petitioner of its “day in court.” *See Vizio, Inc. v. Maxell, Ltd.*, IPR2022-01459, Paper 7 at 21-22 (PTAB Apr. 11, 2023) (“Because Petitioner is the litigation defendant, factor 5 weighs in favor of discretionary denial”); *Nokia of Am. Corp. v. Godo Kaisha IP Bridge*, IPR2022-00755, Paper 9 at 13 (PTAB Nov. 2, 2022) (holding that Factor 5 weighed for discretionary denial where Petitioner was defendant in parallel proceeding).

That other defendants are *also* litigating the petitioned issues at the district court further suggests denial. *Fintiv* holds that, “[e]ven when a petitioner is unrelated to a defendant, however, if the issues are the same as, or substantially similar to, those ... about to be litigated, or other circumstances weigh against redoing the work of another tribunal, the Board may, nonetheless, exercise the authority to deny institution.” *Fintiv*, IPR2020-00019, Paper 11 at 14. Because of this consideration, *Fintiv* goes on to require that “[a]n unrelated petitioner ... discuss why addressing

the same or substantially the same issues would not be duplicative of the [other] case even if the petition is brought by a different party.” *Id.*

Petitioner offers no such explanation here. That other defendants are currently litigating the same issues raised in the petition raises serious concerns of duplicative work by adjudicators, and any public-policy concerns regarding patent validity will be thoroughly addressed by the parallel litigation. Thus, that *other* defendants in the district court proceedings are *also* litigating the petitioned grounds weighs yet further for discretionary denial.

F. *Fintiv* Factor 6: The Petition Does Not Present “Compelling Evidence of Unpatentability.”

Factor 6—the merits of the Petition—also counsels for denial because, as detailed below, the petition fails to show that Petitioner is more likely than not to prevail on any challenged claim. *See* 35 U.S.C. § 324(a). Indeed, the Board has *already rejected* the Petition’s arguments when Sophos raised them against the ’869 Patent’s parent in an earlier petition. *See Sophos IPR Denial*. The reasoning in that denial applies equally here: the combination of Ideses and Mori does not render the claims obvious because it does not teach or suggest turning features on or off *at the classification stage*, as recited. Because the petition is not more likely than not to succeed, it may not be instituted.

Even if the Petition were more likely than not to succeed (it is not), it certainly would not pass the higher threshold of a “compelling” unpatentability challenge that could preclude discretionary denial. Guidance at 4. “[T]he compelling merits standard is a higher standard than the standard for institution set by statute.” *Commscope Techs. v. Dali Wireless, Inc.*, IPR 2022-01242, Paper 23, at 3 (PTAB Feb. 27, 2023). That higher standard requires that the Petition is “highly likely” to succeed on the merits. *Id.* at 4. The opposite is true here.

* * *

In sum, all six factors weigh for discretionary denial. Accordingly, Patent Owner respectfully requests that the Board exercise its discretion to deny institution.

III. THE BOARD SHOULD DENY INSTITUTION BECAUSE PETITIONER IS NOT LIKELY TO SUCCEED.

The Board should deny institution because the petition fails to show it is more likely than not to prevail on any claim. The petition presents six grounds:

Ground	Claims	Argument
1	1-59	§ 103(a): Ideses made the claims obvious.
2	1-6, 8-24, 26-27, 29-30, 32-38, 40-41, 43-44, 46-52, 54-55, and 57-58	§ 103(a): Mori made the claims obvious
3	1-59	§ 103(a): The combination of Ideses and Mori made the claims obvious.
4	1-59	§ 112(b): The phrase “the static data points” renders the claims indefinite.

Ground	Claims	Argument
5	8, 14, 17, 29, 43, and 57	§ 112(b): The idea of “character strings” being “within a predetermined range” renders the claims indefinite.
6	8, 14, 17, 29, 43, and 57	§ 112(b): The patent provides insufficient written description of “character strings” being “within a predetermined range.”

The obviousness grounds (1-3) are unlikely to succeed because—as *the Board has already decided*—none of the art on which those grounds rely (alone or in combination) discloses or suggests a classifier that selectively turns features on or off *at the classification stage*, as claimed.

The Section 112 grounds are also unlikely to succeed. Ground 4 is unlikely to succeed because a POSITA would understand with reasonable certainty that “the static data points” refers to the “static data points” recited just words earlier in the same sentence. And Grounds 5 and 6 are unlikely to succeed because the claims do not require “character strings” to be “within a predetermined range,” and even if they did, a POSITA would understand what that would have meant, and that the inventors possessed that idea at the time they filed the application.⁵

⁵ Because this is a preliminary response, Open Text does not address every limitation of every claim. But this does not constitute an admission that those limitations not

A. Background

1. The State of the Art

Every day, thousands of new executable files (“executables”) are created and distributed across computer networks. (EX-1001 1:18-19.) It is the job of computer security software to quickly recognize which of those executables is malicious or otherwise unwanted, and to prevent execution of them. (*Id.* 1:18-24.)

One conventional approach is to perform “static analysis,” whereby the security software analyzes the executable’s code to identify known malicious code blocks within the executable. (*Id.* 2:23-24.) That conventional approach had two drawbacks: (1) it could not identify malicious files that it had not seen before, and (2) it was slow at evaluating encrypted files because it required the security software to first decrypt and disassemble the code.

One solution to the first problem (*i.e.*, identifying malicious code that has not been seen before) is to use a “machine learning classifier.” (*Id.* 1:27-48.) Machine learning classifiers are software programs that are “trained” to classify new, unknown executables by studying a large group of *known* executables (*e.g.*, “known

discussed are disclosed or rendered obvious by the cited references, and Open Text reserves the right to argue that they are not should the Board institute trial.

malicious executable files, known benign executable files, and [known] potentially unwanted executable files”). (*Id.*) During a “training stage,” the system studies known executables to identify features or combinations of features that are correlated with each category of executable (*e.g.*, malicious, benign, unwanted, etc.). Based on these identified features, the system builds a “classifier”—*i.e.*, a predictive model that accepts a set of values for the features as input and outputs a predicted classification. The model can then be used to classify previously unseen executables: the computer security software simply extracts from the previously unseen executable values for the features used by the classifier, inputs those values for the features into the classifier, and thus obtains a predicted classification for the executable. These are the steps of the “classification stage.” (EX- 2001 ¶¶ 27-33.)

A classifier is typically trained by studying a set of training samples (known executables) to identify a set of features that are useful for classifying future executables. From each executable, the system extracts values for the identified features and stores them in a “feature vector” representing the executable. The system then finds a mathematical function that best separates the feature vectors into respective categories. To classify a new sample, the system creates a feature vector representing the new sample—*i.e.*, the system analyzes the new sample to extract values for the preidentified set of features (a process known as “feature extraction”) and uses those

values to create a feature vector representing the new sample—and it inputs the feature vector into the classifier for classification. (EX- 2001 ¶¶ 26-31.)

Because the computational complexity (and therefore the required computational time and resources) grows with the number of features a classifier has been trained to consider, the classifier training phase may sometimes include a process known as “feature selection” or “feature pruning.” Feature selection is the process of identifying the features that are most effective and/or computationally inexpensive, and configuring the classifier to consider only those features. Feature selection at the training stage reduces the computational burden at the classification stage because the classifier will have to extract fewer features and/or perform less complex mathematical computations. (EX-2001 ¶ 25.)

2. The '869 Patent

The '869 patent improves upon the prior art by introducing an executable classification technique that performs feature selection at the *classification stage, based on data points in the executable being classified*, rather than only at the training stage. Performing feature selection during the classification stage allows the system to dynamically streamline its performance based on the executable being classified rather than always determining values for, and using, all features identified at the training stage.

The invention achieves feature selection during the classification stage by turning features “on” or “off” during that stage based on the content of the executable being evaluated. (EX-1001 7:43-8:7; 12:61-13:11.) For example, the classifier may consider certain features (at the classification stage) only if the size of the executable being evaluated is above a given threshold or only if the executable includes a certain type of content (*e.g.*, legal copyright information). (*Id.* 7:43-8:7.)

To decide which features should be turned on or off, the system analyzes “static data points” in the executable. The specification teaches that a “static data point” may be a “point of reference,” such as the executable’s file size, section names, legal information, and so forth. (*See* EX-1001 5:61-6:44, 8:39-43.)

The specification provides many examples of static data points (*see id.* Table 1.1, 8:39-43), and a person of ordinary skill would recognize that provided examples may correspond to fields defined by the “Portable Executable” (or “PE”) file format. (EX-2001 ¶¶ 29-30; EX-1001 3:43-54 (discussing the extraction of static data points from PE files as the preferred but not exclusive embodiment), *id.* 4:29-31 (describing extraction of static data points from PE files).) In such embodiments, a static data point may include a field name (*e.g.*, a predefined character string such as “File Size”) and a field value (*e.g.*, a number such as 4982784). (EX-1001 8:39.)

The claimed system can turn features on or off *during the classification stage*

based on the values of the static data points extracted from the executable being classified. (*Id.* 5:61-6:44, 8:39-43, 7:43-8:7 (describing selectively turning on or off features during “evaluation”).) For example, if the value of the “File Size” static data point is greater than a certain numerical threshold, then the classifier may determine values for fewer or additional features. (*Id.*) The set of features ultimately extracted from the executable being classified may be used to create one or more feature vectors that are input into the appropriate classifier. (*Id.* 8:43-9:26.)

Turning a feature “off” at the classification stage improves classifier performance because it allows the classifier to not use that feature in its prediction model, and thus, the classifier need not waste time and resources (1) determining a value for that feature, and/or (2) considering the value of the feature in the classification algorithm. (EX-2001 ¶ 25.) This approach contrasts with traditional approaches, where at the classification stage, the classifier had to determine values for every feature predetermined in the training stage and consider those values. (*Id.*)

Claim 1 recites the invented technique, including both the training and classification stages:

1. A computer-implemented method comprising:

- identifying, by a knowledge module, static data points that may be indicative of either a harmful or benign executable file;

associating, by the knowledge module, the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files;

identifying an executable file to be evaluated;

extracting, by the knowledge module, a plurality of static data points from the identified executable file;

generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data,

the training data comprising files known to fit into one of the plurality of categories of files,

wherein one or more features of the feature vector are selectively turned on or off based on whether a value of one or more static data points from the plurality of extracted static data points is within a predetermined range; and

evaluating the feature vector using support vector processing to determine whether the executable file is harmful.

(EX-1001 (claim 1).)

The claimed method begins with a *training* stage in which a “knowledge module” identifies static data points and associates them with categories of executables—*e.g.*, “harmful” or “benign.” The method then continues to a classification stage, which calls for “identifying an executable file to be evaluated,” extracting “static data points from the identified executable,” and “generating a feature vector

from ... [those] static data points.” The feature vector is generated using a classifier that was previously “trained”—*i.e.*, at the training stage—to classify the static data points based on training data comprising files known to fit into one of the ... categories of files.” One or more features of the feature vector are then “turned on or off” based on whether a value of at least one of the static data points extracted from the executable being evaluated is within a predetermined range. For example, the classifier might turn certain features “off” if the “File Size” of the executable is smaller than some predefined threshold. Finally, the classifier classifies the executable by using support vector processing on the resulting feature vector.

Independent claims 10, 15, 18, 32, and 46 all recite systems and methods that turn features of a feature vector on or off at the classification stage based on the value of static data points extracted from the executable being evaluated. Claims 10 and 15 require “providing the generated feature vector to one or more support vector machines to build a probabilistic model that indicates whether the executable file fits into one of the categories of files.” Claims 18, 32, and 46 require “evaluating the feature vector using a machine learning model to determine whether the executable fits into one of the categories of files.”

3. The Board's Rejection of Sophos's Earlier Petition Against the Parent '844 Patent

On February 7, 2023, Sophos petitioned for IPR of the '844 patent, raising the same prior art arguments presented here in Grounds 1-3—specifically, that Ideses, Mori, or the combination thereof rendered the claims of the '844 patent obvious. *See Sophos*, IPR2023-00528, Paper 1.

On August 14, 2023, the Board refused to institute trial on Sophos's petition because Sophos had not “demonstrated a reasonable likelihood that it will prevail.” *Sophos IPR Denial* at 2. Specifically, the Board found that the claims of the '844 patent are “directed to the classification stage” because they (1) “require ‘extracting a plurality of static data points from an executable file’ and ‘generating a feature vector from [that] plurality of static data points *using a classifier*,’” and also because (2) they “recite that the feature vector generated from the static data points that were extracted from the executable file is used ‘to determine whether the executable file is harmful,’ confirming that the claimed method is about classification of an executable file, not creation of a classifier.” (*Id.* at 18.) The Board reasoned that neither Ideses nor Mori teaches, suggests, or renders obvious the idea of feature selection at the classification stage.

The '869 patent is a continuation of the '844 patent. Like the '844 patent, the '869 patent claims feature selection at the classification stage.

4. Claim Construction

The parties agree that no claim terms need be construed for this petition, and all claim terms should carry their plain and ordinary meaning in light of the intrinsic record. (Petition 7-10.)

As referenced above, the parties are involved in parallel litigation where the district court has issued several interlocutory claim construction orders relevant here (EX-2003 17-20; *id.* 19.) The Board's regulations require that the Board consider those constructions. 37 C.F.R. § 42.100(b).

In analyzing the '869 patent, the district court construed two terms:

- **“the static data points”** The district court held that “static data points” is not indefinite, and that the phrase takes its plain and ordinary meaning. (EX-2003 19.) In reaching that conclusion, the district court rejected the same indefinite arguments Petitioner present in the present Petition. (*Compare* EX-1008 52-54 with Petition 87-88.)
- **“within a predetermined range”** The district court held that the phrase “within a predetermined range” carries its plain and ordinary meaning, and it defined “range” as a “numerical range or a subset of a ‘particular scale’ with at least a lower limit or an upper limit, but not necessarily both.” (EX-2003 19.)

In analyzing the parent '844 patent, the district court construed four additional terms that are relevant here because they appear also in the '869 patent:

- “character strings” being “within a predetermined range”** The district court held that it was not indefinite for a “character string” to be “within a predetermined range,” as recited in claim 17, and that this phrase should be given its plain and ordinary meaning. (EX-1007 18.)
- **“executable file”** The district court held that the phrase “executable file” carries its plain and ordinary meaning. (*Id.* 17.) The court rejected a proposal that would have required “executable files” to be “encrypted or packed.” (*Id.*)
 - **“support vector processing”** The district court held that “support vector processing” carries its plain and ordinary meaning, and that this meaning is “processing that involves a support vector.” (*Id.* 18) The court rejected arguments that “support vector processing” is limited to “processing with a support vector machine.” (*Id.*)
 - **“at least one processor”** The district court held that this phrase has its plain and ordinary meaning, and it rejected the argument that the phrase is a means-plus-function limitation under § 112(f). (*Id.* 19.)

5. The Level of Ordinary Skill

For purposes of this proceeding, Patent Owner does not dispute Petitioner's articulation of the level of ordinary skill. (Petition 10.)

B. Ground 1: Ideses Fails to Render Any of the Challenged Claims Obvious

Petitioner fails to show it is more likely than not that Ideses renders independent claims 1, 10, 15, 18, 32, and 46 obvious. Petitioner therefore fails to establish that Ideses renders any of the dependent claims obvious.

As the Board has already found when refusing to institute IPR on the parent '844 patent (*Sophos IPR Denial* at 18-19), Ideses discloses the traditional idea of selecting features at the *training* stage, *based on how effective* those features are and/or how computationally expensive they are to extract (EX-1005 [0002], [0060-0061]), and that idea differs from that of selecting features at the *classification* stage, *based on static data points extracted from the executable being classified*, as recited in the challenged claims. Because Ideses fails to disclose turning features on or off at the classification stage, it does not render the challenged claims obvious. The Board should reject Petitioner's argument to the contrary as "the same or substantially the same prior art or arguments previously [] presented to the Office." § 325(d).

1. Overview of Ideses

Ideses is directed to generating classifiers that are efficient enough to be executed on resource-constrained mobile devices. (EX-1005 [0002], [0060-0061].)

Ideses observes that executing classifiers (*i.e.*, performing the classification stage) on a mobile device is problematic because “mobile devices may be resource limited, [and therefore] unable to extract the full set of features [from an executable] during run-time, and/or unable to perform the extraction within a reasonable amount of time.” (*Id.* [0060].)

The solution Ideses proposes is for a server to identify an optimal set of features *at the training stage* “so that [the subsequent] classification [stage] may take place [on a mobile device] during run-time with the resource availability of the mobile device.” (*Id.* [0061].) Ideses's training stage (as performed by the server) is shown in Figure 1, and the classification stage (as performed by the mobile device) is shown in Figure 2.

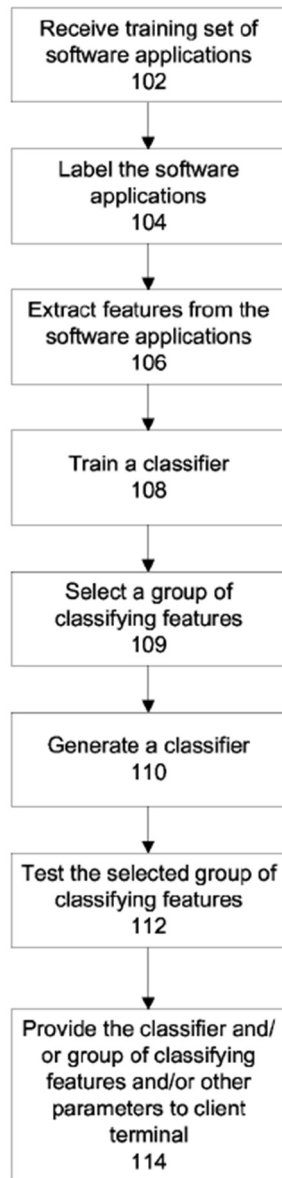


Fig. 1 (Training Stage)

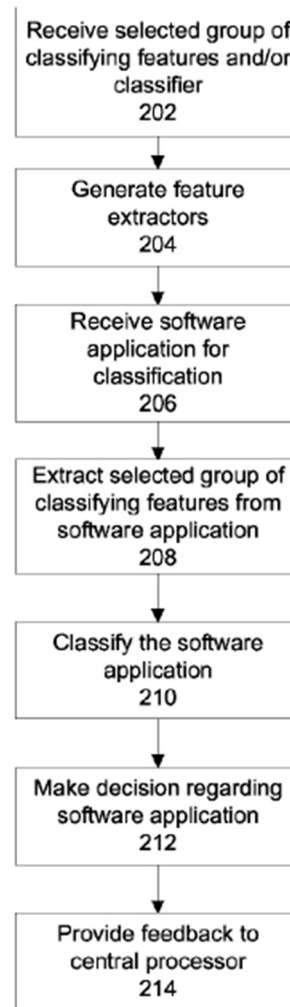


Fig. 2 (Classification Stage)

(EX-1005, FIGS. 1, 2 (annotated).)

The training stage (Fig. 1) starts with the traditional steps of (1) receiving a training set of software applications (EX-1005 [0080]), (2) labeling each application in the training set with a classification type, such as “desired” or “undesired” (*id.*

[0081]-[0085]), (3) extracting features from each application (*id.* [0086]-[0089]), and (4) using the features and labels to train a classifier, such as an SVM (*id.* [0094]-[0096]).

The training stage includes feature selection, *i.e.*, the process of selecting the features that the trained classifier will be trained to consider (step 109). Ideses teaches that “the selection [of features] is based on identifying a ***classification effectiveness rank*** for each of the features” and/or “***resource requirements*** for each of the features.” (*Id.* [0099] (emphasis added).) In other words, features are selected at the training stage “based on the tradeoff of improving classification ability, while reducing CPU requirements and/or ... memory usage.” (*Id.* [0113].) In this way, “the group of classifying features is selected to allow calculation and/or extraction of the features during run-time on the mobile device.” (*Id.* [0103].)

The classifier is then “trained based on the selected group of classifying features.” (*Id.* [0116].) The trained classifier is tested for performance and accuracy (step 112), and it is retrained (potentially with different features) if the performance is unsatisfactory. (*Id.* [0118]-[0119].) At step 114, the final classifier is sent to the mobile device for use in classifying new executables. (*Id.* [0120].)

Figure 2 illustrates the classification stage, as executed on the mobile device—*i.e.*, “execution of a classifier based on a pruned feature list.” (*Id.* [0123].) The

mobile device: (1) receives the classifier and pruned feature list from the server (step 202); (2) generates or identifies a feature extractor for the pruned feature list (step 204); (3) receives a new executable to evaluate (step 206); (4) uses the extractors to extract the pruned feature list from the new executable (step 208); (5) inputs the extracted features into the classifier to classify the new executable (step 210); (6) uses the output of the classifier to decide whether to execute the new executable (step 212); and (7) reports the outcome back to the server (step 214).

2. Ideses Fails to Render Any of the Independent Claims Obvious Because It Does Not Teach or Suggest Turning Features On or Off at the Classification Stage Based on the Executable Being Classified

As the summaries above make plain, the idea recited in the independent claims of the '869 patent differs from that disclosed in Ideses. The idea in Ideses is that of traditional feature selection: at the *training* stage, the system selects the features that the classifier will be trained to consider, and it does so based on how effective and/or computationally expensive the features will be to extract and evaluate at the classification stage. Thus, at the later classification stage, the set of features that must be extracted and considered has already been predetermined, such that there is *no feature selection at the classification stage*. In contrast, the idea claimed in the '869 patent is to turn features on or off at the *classification* stage, based on the executable being classified at *that* stage. Ideses does not teach or suggest that approach.

The language of the independent claims clearly requires turning features on or off at the *classification stage* because the generated feature vector—whose features are “turned on or off”—is not generated until that stage. For example, claim 1 recites (1) “identifying an executable to be evaluated” (*i.e.*, classified at the classification stage), (2) “extracting ... static data points from the identified executable,” and (3) “generating a feature vector from the plurality of static data points,” “wherein one or more features of the feature vector are ... turned on or off based on ... [the] extracted static data points.” (EX-1001, Cl. 1.) Because the recited turning of features “on or off” is performed on the feature vector created at the classification stage, that turning on or off necessarily happens at the classification stage too.

Confirming this timing, the claims further recite that the feature vector whose features are being turned on or off is generated “using a classifier [that has already been] ***trained***”—past tense. (*Id.* at 16:22.). Independent claims 18, 32, and 46 further recite that the feature vector is used to classify the identified executable (“evaluat[ed] ... to determine whether the executable file fits into one of the categories of files”), meaning that the identified executable is the one being classified at the classification stage. (*Id.* at 18:12-14.) Claim 20 further emphasizes the distinction between the recited training and classification stages by separately reciting them as two different

sets of operations, each set separately recited as being performed by “at least one processor.” (*Id.* at 19:55-20:21.)

In sum, because the “executable” for which the feature vector is generated is the executable being classified at the classification stage, the requirement of turning features of that feature vector “on or off based on ... [the] extracted static data points” from that executable must necessarily happen at the classification stage too. Thus, the challenged claims all require turning features on or off *at the classification stage* based on the executable being classified.

The benefit of turning features on or off at the classification stage is that classifier performance can be optimized *based on the executable being classified*. Unlike the prior art, the classifier is not locked into evaluating the full set of features that was predetermined at the training stage without regard to the executable being classified at the classification stage. (EX-2001 ¶ 25.)

Ideses fails to disclose turning features on or off at the classification stage based on the executable being classified. In contrast, Ideses teaches feature selection at the *training stage* before the executable being classified is ever seen.

To argue that Ideses teaches turning features on and off, Petitioner points to “the method depicted in *Figure 1*” of Ideses. (Petition 33.) But Figure 1 illustrates the *training* stage of Ideses, *i.e.*, the steps starting from receiving the training set

(step 102) and ending in ultimately “generat[ing] a classifier” (step 110) and “provid[ing] the classifier ... to [the] client terminal” (step 114). It is that terminal that later performs the steps of the classification stage, *as illustrated in Figure 2*. (See EX-2001 ¶¶ 63-67.)

The feature selection shown in Ideses Figure 1 (step 109) therefore occurs during the *training* stage, which is *before* the system “receive[s] [the] software application for classification” (which occurs at step 206 of Figure 2). Because the feature selection in Figure 1 (step 109) happens *before* the system receives the “software application for classification” (step 206), that feature selection *cannot* be based on the “software application for classification” (step 206), as required by the challenged claims. (See EX-2001 ¶¶ 64-67.)

In Ideses, the classification stage is shown in Figure 2. (EX-1005 [0123] (“Fig. 2 ... is a flow chart of a method of run-time execution of a classifier.”).) The illustrated process in Figure 2 does not include anything that Petitioner has identified as turning features on or off. (Petition 33-35.) Accordingly, Petitioner has not shown (and cannot show) that Ideses disclosed or suggested a malware classifier that, at the classification stage, generates a feature vector from static data points using a classifier already trained to classify those data points and turns features “on and off based on whether a value of one or more static data points from the [executable being

classified] ... is within a predetermined range,” as recited by the challenged independent claims.

3. Ideses Fails to Render Any of the Dependent Claims Obvious

Because Petitioner cannot show that it is more likely than not to succeed on any of the *independent* claims, it cannot meet its burden for the challenged dependent claims. *Twinstrand Biosciences, Inc. v. Guardant Health, Inc.*, IPR2022-01115, Paper 14, 19 (PTAB Dec. 30, 2022). Petitioner has thus failed to show it is more likely than not to prevail on Ground 1, with respect to any of the challenged claims.

C. Ground 2: Mori Fails to Render Any of the Challenged Claims Obvious

Petitioner argues in Ground 2 that Mori renders all claims obvious, except for dependent claims 7, 25, 28, 31, 39, 42, 45, 53 56, and 59. That is wrong because Mori has similar failings to Ideses. As the Board found when refusing to institute an IPR of the '844 patent (*Sophos IPR Denial* at 20-21), Mori teaches feature selection only *at the training stage*, not at the classification stage based on the executable being classified. Here too then, Petitioner is simply repeating “the same or substantially the same prior art or arguments previously [] presented to the Office.” § 325(d).

1. Overview of Mori⁶

Like Ideses, Mori discloses a technique for training a machine learning classifier (*e.g.*, an SVM) to detect malware. Mori's approach is to "focus[] on *readable strings*" in an executable, *i.e.*, words in an executable that appear in a predefined dictionary (or, alternatively, those that the system can find on the internet). (EX-1006 [0023] (emphasis added), *see also id.* [0032], [0066].)

Mori's disclosure is devoted almost entirely to describing the training stage, during which the system analyzes a training set of executables to identify a feature set to use for classification and trains a classifier (*e.g.*, an SVM) to classify new executables based on values of the features in that set for the new executable. (*Id.* [0010], [0029].)

The system begins by identifying all readable words that appear in the executables in the training set. (*Id.* [0030], [0052]-[0053]) (teaching that the feature vectors are created "[u]sing each word ... in each file"). The system then creates a feature vector for each executable. Each feature vector represents *M* features, where

⁶ The following analysis is based on the translation of Mori Petitioner supplied. Patent Owner does not concede Petitioner's translation is accurate and reserves the right to obtain its own certified translation.

each feature corresponds to a respective one of the readable words identified across all executables in the training set. (*Id.*)

For each executable, the system sets the value of each of the *M* features in the executable's feature vector to reflect the occurrence, frequency, or importance (in the executable) of the readable word corresponding to that feature. (EX-1006 [0030].) For example, the value of a given feature could be set to 1 or 0 based on "whether a word [corresponding to that feature] existed [in the executable] or not." (*Id.*) Alternatively, the value of a feature could reflect the "frequency" or "normalized frequency" with which the corresponding word appears in the executable or a "tf-idf value" for the word (*i.e.*, the word's frequency in the executable compared to its frequency in the other executables in the training set). (*Id.*)

In all of Mori's disclosed embodiments, the value for a feature is 0 if the readable word corresponding to that feature does not appear in the executable. (EX-2001 ¶¶ 52-53.) For example, in embodiments where the value is set to 1 or 0 based on "whether a word existed or not," the absence of a given word would be noted by setting the corresponding feature in the executable's feature vector to 0. (EX-1006 [0030].) Similarly, as Petitioner observes, embodiments where the value of each feature is a tf-idf value for the corresponding word, the absence of the corresponding word in an executable would likewise be noted with a value of 0 for the

corresponding feature in the executable's feature vector "because the tf portion [*i.e.*, the word's frequency] is 0." (Petition 69.) Thus, in all disclosed embodiments, the absence of a given readable word from an executable is noted in the executable's feature vector by setting the value of the feature corresponding to that readable word to 0. In other words, the feature for a missing word is not omitted from the feature vector, but rather, the value corresponding to that feature is set to 0.

Once the system creates a feature vector for each executable in the training set, it uses the feature vectors to train a classifier (*e.g.*, an SVM), and that classifier can be used at the classification stage to classify new executables.

Mori describes the classification stage only in passing. (*See* EX-1006 [0054]-[0055]; *see also id.* [0034].) At that stage, the system analyzes an executable to determine values for the *M* features defined in the training stage, *i.e.*, the occurrence or frequency with which the *M* words predefined in the training stage appear in the executable. The system places those values in a feature vector for the executable and inputs that feature vector into the classifier for classification. (*Id.* [0054].)

2. Mori Fails to Render Any of the Independent Claims Obvious Because It Does Not Teach or Suggest Turning Features On or Off at the Classification Stage Based on the Executable Being Classified

Mori does not render any of the independent claims obvious because, like *Ideses*, Mori does not teach or suggest turning features "on or off" at the

classification stage, based on the executable being classified. Instead, Mori's system is no different from prior art classification stages, wherein the classifier extracts and analyzes the same set of features that was predefined at the training stage. Because the choice of features that Mori extracts and considers at the classification stage does not depend on the content of the executable being evaluated, Mori does not teach, suggest, or render obvious the claimed idea of turning features "on or off based on ... a value of ... one or more static data points" extracted from the executable being classified at the classification stage.

As discussed above, the invention recited in claim 1 improves classifier performance by allowing the system to decide which features are "turned on or off" *at the classification stage, based on the executable being classified*. Again, the independent claims all recite: (1) "identifying an executable to be evaluated" (*i.e.*, classified at the classification stage), (2) "extracting ... static data points from the identified executable," and (3) "generating a feature vector from the plurality of static data points," "wherein one or more features of the feature vector are ... turned on or off based on ... [the] extracted static data points." Thus, the '869 patent's system can determine which features identified at the training stage will be "turned on or off" at the classification stage based on the executable being classified at that stage. (EX-1001 claim 1.) For example, the "classifier may turn on/off features for

evaluation based on whether [] the size of the file [being classified] is within a certain range.” (*Id.* 7:45-52.)

Mori's system does not disclose or suggest turning features on or off at the classification stage. Instead, Mori's system determines a set of features at the *training stage*—*i.e.*, one corresponding to each readable word identified in the training set's executables—and trains a classifier to classify executables based on that set of features. (EX-2001 ¶ 73.) Then, at the classification stage, the classifier extracts from the executable the values for the same set of features determined at the training stage, setting the value of each feature to describe the occurrence or frequency in the executable of the readable word corresponding to the feature. (EX-2001 ¶¶ 74-76.) The resulting feature vector therefore conveys values for all features used to train the classifier in the training stage. (EX-2001 ¶ 77.) The full feature vector is then input into the classifier, which considers the values of all the features in the feature vector to make the classification. (*Id.* ¶¶ 76-77.) Thus, Mori's classifier does not turn any features “on or off” at the classification stage. Instead, it always extracts and considers the same set of features predetermined at the training stage.

Petitioner articulates two theories for how Mori's classifier turns a feature on or off (Petition 66-70), but neither is correct.

First, Petitioner argues that Mori's classifier turns a feature "off" when it sets the value of that feature to 0, as would happen when the readable word corresponding to that feature does not appear in the executable. (Petition 67.) That is wrong because a feature whose value is 0 is still a feature whose value is determined and considered by the classifier during the classification process. (EX-2001 ¶ 77-78.) The value of 0 simply indicates that the readable word corresponding to that feature does not appear in the executable. (EX-2001 ¶ 78.) Mori's classifier considers that information when it classifies the executable no less than it considers the information conveyed by non-zero values. (*Id.*) In either case, the system still spends time and resources extracting a value for the feature, and the system still feeds the extracted value into the classifier for use in the classification process. (*Id.*) Thus, a feature whose value is set to 0 is not turned "off" during classification within the meaning of the claims. Rather, it (and its value) are still used during classification.

Second, Petitioner argues that Mori turns a feature "off" by *omitting* the feature from the feature vector if the feature corresponds to a word that is absent from the executable being evaluated (*i.e.*, if the feature has a value of 0). (Petition 69 ("if a word is not present, the 'static data point' for that word is 'turned off' since it is not included in the generated feature vector for that file".)) Petitioner seems to base this argument on the example feature vectors partially displayed in Mori's Table 5,

where each feature vector (one in each row) appears to include entries only for features with non-zero values. (Petition 68.)

Petitioner's argument contradicts Mori's full disclosure, which teaches that features with 0-values *are* included in the extracted feature vectors. (EX-1006 [0030].) Mori discloses that the value of a feature can "be defined as whether a word existed or not by assigning it a '0' or '1'," or by various measures of frequency that (in every disclosed embodiment) have a value of 0 when the corresponding readable word does not appear in the executable. (*Id.*; EX-2001 ¶¶ 80-81.)

Moreover, Mori provides no description of a classifier that *can* operate on feature vectors that comprise different sets of features, let alone of *how* such a classifier would work. If Petitioner were right that a single instance of Mori's classifier could classify one executable based on one set of features and another executable based on a different set of features without being retrained (*i.e.*, that it can turn features on or off at the classification stage), one would expect Mori to describe that functionality and how the classifier accomplishes it. But Mori includes no such disclosures. The absence of those disclosures confirms that Mori does not disclose a classifier that can turn features on or off at the classification stage, as recited.

Mori's disclosure in Table 5 (which does not show vectors with zero-value features) is not inconsistent with Mori's disclosure that absent readable words are

represented by 0-values (at [0030]) or with Mori's failure to disclose a classifier that can operate on feature vectors with different sets of features. From these teachings, one of ordinary skill could understand that, at the classification stage, Mori's classifier simply assumes (or inserts) a value of 0 for each feature not shown in the feature vectors of Table 5. (EX-1006, at [0030].) For example, using a common technique known as "vector padding," the classifier could reformat the feature vectors shown in Table 5 to include a 0 value for each missing feature (as described at [0030]), such that the reformatted feature vector would include values for all *M* features that the classifier was trained to accept. (EX-2001 ¶ 80-82.) Consequently, notwithstanding the partial feature vectors shown in Table 5, Mori's classifier extracts and considers values for all *M* features that had been predetermined at the training stage, meaning the system would not turn any of those features on or off at the classification stage, as challenged claim 1 requires.

Thus, Mori does not teach turning features "on or off" at the classification stage, and it discloses no mechanism for doing so. Instead, Mori's classifier considers values for all *M* features used to train the classifier at the training stage, and the disclosure of partial feature vectors in Table 5 is not to the contrary. Mori cannot render obvious the claimed idea of turning features "on or off based on" anything in the executable being classified, let alone "based on whether a value of one or more

... extracted static data points [from the executable] is within a predetermined range,” as recited in claim 1.

3. Mori Fails to Render Any of the Dependent Claims Obvious

Because Petitioner cannot show that it is more likely than not to succeed on any of the *independent* claims, it cannot meet its burden for the challenged dependent claims. *Twinstrand Biosciences*, IPR2022-01115, Paper 14, 19. Petitioner has thus failed to show it is more likely than not to prevail on Ground 1, with respect to any of the challenged claims.

D. Ground 3: The Combination of Ideses and Mori Does Not Render Claims 1-59 Obvious

In Ground 3, Petitioner argues that the combination of Ideses and Mori rendered the challenged claims obvious (Petition 83-86) because, “to the extent ... Ideses does not sufficiently teach or suggest any element of any of the challenged independent claims, Mori discloses these features.” (*Id.* at 83). That argument fails because neither Ideses nor Mori teach, suggest, or render obvious the idea of turning features on or off at the classification stage.

Critically, Petitioner does not argue that the combination of Ideses and Mori teaches anything more about the missing “on or off” feature than do the references in isolation. Rather, Petitioner argues only that the combination would have rendered obvious two limitations that appear only in certain *dependent* claims: (1) analyzing

the executable “without decrypting or unpacking [it]” (dependent claims 2, 11, 16, 19, 33, 47), and (2) extracting “predefined character strings” (dependent claims 8, 14, 17, 29, 43, 57). Because Petitioner presents no independent argument that the *combination* would have rendered it obvious to modify Ideses’s system to turn features on or off at the classification stage, Ground 3 must fall with Ground 1.

Petitioner’s arguments regarding the two dependent-claim limitations are also flawed because they fail to explain either (1) the proposed combination or (2) why a POSITA would have had a reasonable expectation that the combination would succeed. But because Ground 3 must fall with Ground 1, the Board need not reach those issues. Patent Owner reserves the right to raise those arguments in its full Response, should the Board institute review.

E. Ground 4: The Phrase “The Static Data Points” Does Not Render Claims 1-59 Indefinite

In Ground 4, Petitioner repeats a claim construction argument already rejected by the district court in the parallel litigation. (Petition 8.) Petitioner’s argument is that the claims are indefinite because a POSITA would not know with reasonable certainty whether “the static data points” analyzed during the classification stage are (1) those extracted from the executable being evaluated at the classification stage or instead (2) those extracted earlier from other executables at the training stage. (Petition 86-88.)

This argument fails because the claim language, read in context and in light of the intrinsic record, plainly conveys that “the static data points” analyzed during the classification stage are the “plurality of static data points” extracted from the executable being evaluated during the classification stage. Not only are the two phrases recited mere words apart in the same clause, but the entire point of the classification stage is to evaluate the “executable file to be evaluated” at that stage.

Claims commonly use definite articles (like “the”) to indicate that an element refers to a previously-recited element (*i.e.*, an antecedent). Where two possible antecedents exist, courts have typically identified the correct antecedent by analyzing the context of the claim language, the specification, and the prosecution history. *See, e.g., Altera Corp. v. PACT XPP Techs.*, AG, No. 14-CV-02868-JD, 2015 WL 4999952, at *14 (N.D. Cal. Aug. 21, 2015) (“the unit” did not render indefinite a claim that earlier recited two instances of “unit”); *Blast Motion, Inc. v. Zepp Labs, Inc.*, No. 15-CV-700 JLS (NLS), 2017 WL 476428, at *6 (S.D. Cal. Feb. 6, 2017) (“said data” did not render indefinite a claim that earlier recited two instances of “data”); *InfoGation Corp. v. ZTE Corp.*, No. 16-CV-01901-H-JLB, 2017 WL 1821402, at *13 (S.D. Cal. May 5, 2017) (“said optimal route” did not render indefinite a claim that earlier recited “optimal routes”); *Intell. Ventures II LLC v. BITCO Gen. Ins. Corp.*, No. 615CV59LEADCASE, 2016 WL 125594, at *16 (E.D. Tex.

Jan. 11, 2016) (“the particular user” did not render indefinite claim that earlier referenced multiple “users”); *Tuna Processors, Inc. v. Hawaii Int’l Seafood, Inc.*, 327 F. App’x 204, 209 (Fed. Cir. 2009) (“the smoke” did not render indefinite a claim that earlier made multiple references to “smoke”). That approach accords with the Supreme Court’s order to evaluate definiteness “in light of the specification and prosecution history.” *Nautilus Inc. v. Biosig Instruments, Inc.*, 572 U.S. 898, 910 (2014).

We start with the claim language. Here, the language alone strongly suggests that “the static data points” refers to “the plurality of static data points” of the classification stage because the two phrases appear in the same clause. The disputed clause is “generating a feature vector from the **plurality of static data points** using a classifier trained to classify **the static data points**.” In this context, it would be unreasonable to conclude that “the static data points” do *not* refer to the “plurality of static data points” recited only six words earlier in the same clause, and instead, refer to the “identified static data points” last recited three clauses earlier. As a matter of plain English, “the static data points” refers to the textually adjacent “plurality of static data points.”

The correctness of the plain-English reading is reinforced when the claim is read from the perspective of a POSITA. A POSITA would understand that the

purpose of “train[ing]” a “classifier” is to enable the classifier to evaluate the “executable to be evaluated” at the classification stage. Thus, when the claims recite a “classifier trained to classify the static data points,” a POSITA would naturally understand that the classifier was trained to classify “static data points” extracted from the “executable to be evaluated.” (EX-2001 ¶¶ 94-96).

The intrinsic record further confirms this understanding. During prosecution, the applicant expressly identified the portions of the specification that disclose the disputed “generating” limitation. (EX-1002 79, 121, 132.) Those portions (EX-1001 12:43-13:11) teach that the recited “classifier” is “the learning classifier trained by the knowledge database.” (EX-1001 12:48-51.). The specification explains that this classifier is “train[ed] ... to evaluate threats in executable files” (*Id.* 11:60-12:10.) This disclosure confirms that the recited “classifier” is “trained to classify the static data points” in the “executable file to be evaluated” (*i.e.*, the “plurality of static data points”). Indeed, that is the entire point of training the classifier.

Similarly, other parts of the specification are express that the claimed system applies the classifier “to the plurality of static data points extracted from the executable file.” (*Id.* at 1:36-38.) A POSITA would thus understand that the recited “classifier” is “trained” to classify “the plurality of static data points” extracted from the “executable to be evaluated” during the classification stage.

Moreover, the definiteness conclusion would not change even if “the static data points” could reasonably be understood as referring to *both* the training stage and classification stage data points. That is because a POSITA would understand that a learning classifier is “trained to classify” an executable at the classification stage by being trained to classify executables at the training stage. (EX-2001 ¶¶ 94-96.) Thus, there is no difference in claim scope between a reading that maps “the static data points” only to “the plurality of static data points” of the classification stage and one that also maps “the static data points” to the “identified static data points” of the training stage.

F. Ground 5: The “String” Limitations Do Not Make Dependent Claims 8, 14, 17, 29, 43, or 57 Indefinite

In Ground 5, Petitioner repeats another claim construction argument that the district court rejected in the parallel litigation. (Petition 6-7.) Petitioner argues that certain dependent claims are indefinite because they require determining whether “a value of” a static data point is “within a predetermined range” (as recited in the independent claims) and *also* that the static data point “represent [a] predefined character string[]” (as recited in dependent claims 8, 14, 17, 29, 43, and 57). (Petition 88-94). According to Petitioner, that combination is impossible because one cannot determine whether a “character string” is “within a ... range.” (*Id.*)

This argument makes three incorrect assumptions: (1) that the claims require determining whether a “character string” is “within a ... range,” (2) that only numeric values can be “within a ... range,” and (3) that a “character string” cannot be a numeric value.

First, Petitioner is wrong that the claims require determining whether a “character string” is “within a ... range.” The claim language requires that a static data point “represent [a] predefined character string[]” and that “a value of” the static data point is “within a ... range.” But that language does not require that the “value” is the “predefined character string.” The specification illustrates this by disclosing embodiments where a “static data point” *both* represents a predefined character string and has a separate numeric value. (Ex-1001 at 8:39-42, 7:48-52). For example, the specification discloses an embodiment where (1) a static data point “represent[s]” a field called “File Size” (*i.e.*, a “predefined character string”) and (2) has a “value of” of 4982784 (*i.e.*, a numeric value). Thus, the recited “a value of” the static data point (*e.g.*, 4982784) need not be the recited “predefined character string” represented by the static data point (*e.g.*, “File Size”). Petitioner is therefore wrong that the challenged dependent claims require determining whether the “character strings” are within a “range.”

Second, Petitioner is wrong that a character string cannot be “within a predetermined range.” It is simple to determine that “b” is within the range of “a” to “c,” that “February” is within the range of “January” to “March,” and that “warm” is within the range of “hot” to “cold.” One need only choose the appropriate range. That the claims do not specify the range for character strings does not render the claims indefinite. Indeed, the claims do not specify the range for *numeric* values either. Rather, the claims need only avoid “uncertainty” about what constitutes infringement. *Nautilus*, 572 U.S. at 909-10; *see also S3 Inc. v. NVIDIA Corp.*, 259 F.3d 1364, 1369 (Fed. Cir. 2001) (“The purpose of claims is not to explain the technology or how it works, but to state the legal boundaries of the patent grant.”) And here, it would be simple to determine whether an accused product evaluates a “character string” against a “predetermined range.” Consequently, the challenged dependent claims are definite.

Third, Petitioner is wrong that a “character string” cannot be a number. A POSITA would understand that a computer represents “character strings” as numbers and that programming languages normally permit direct conversion of “character strings” to numbers. (EX-2001 ¶¶ 31–33.) The specification confirms that understanding by equating the term “string” with a “byte sequence” (EX-1001 at Table 1.1), which a POSITA would understand is a sequence of 8-bit binary numbers. (EX-

1001 ¶¶ 31–33.) Even Petitioner admits that computers ultimately represent strings as “numbers.” (Petition 92.). Thus, even if the claims do require determining whether a “character string” is “within a ... range” (they do not) and such a determination can be performed only on numbers (also false), there would still be no contradiction because a “character string” can be a number.

G. Ground 6: The “String” Limitations in Dependent Claims 8, 14, 17, 29, 43, and 57 Have Adequate Written Description

Petitioner's Ground 6 is that dependent claims 8, 14, 17, 29, 43, and 57 are invalid under § 112 because there is “no written description in the specification of how character strings can fall within a numeric ‘range.’” (Petition 94.)

This argument fails for the same reasons discussed above in response to Ground 5. The claims do not require that “character strings” fall within a “range” (let alone a “numeric range”). Consequently, there is no need for “written description in the specification of how character strings can fall within a numeric ‘range.’” (*Id.*)

If the Board holds that the claims do require evaluating whether a “character string” falls within a “range,” the specification would still provide adequate written description because it discloses that (1) “ranges of data can be identified for static data points” (EX-1001 at 7:43-52) and that (2) “static data points” can have “string” values (EX-1001 at Table 1.1). That disclosure plainly suggests that “ranges of data can be identified for” “string” values, and thus, that the inventors possessed that idea.

The specification never limits “ranges of data” to numerical ranges. Instead, it describes numerical ranges as only “one instance” of a range (*e.g.*, a file size range). (EX-1001 at 7:48-52). Indeed, the specification provides examples of values that are not necessarily numeric, yet are still readily amenable to “evaluation of whether ... [the value] is within a predetermined range.” For example, the specification discloses that static data points may include “legal information,” which “may ... includ[e]: timestamp data” (*Id.* at 7:54-56.) Based on this disclosure, a POSITA would recognize that the inventors possessed the idea of using a “range of data” for timestamps.

The specification need not elaborate on every type of “range[] of data” to provide written description support. The “written description requirement is met when the specification clearly allows persons of ordinary skill to recognize that the inventor invented what is claimed.” *In re Glob. IP Holdings LLC*, 927 F.3d 1373, 1376–77 (Fed. Cir. 2019) (internal quotation marks omitted). “[T]he level of detail required ... varies” depending on the “existing knowledge in the particular field” and the “predictability of the relevant technology.” *Ariad Pharms., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010). And here, the suggestion to evaluate whether a “string” falls within a “range[] of data” is sufficient to show that the inventors possessed that idea. (EX-2001 ¶¶ 104, 105.)

The challenged dependent claims are therefore supported by sufficient written description, and Ground 6 fails.

H. There are objective indicia of non-obviousness

In determining obviousness of the claimed invention, objective evidence of secondary considerations pertaining to nonobviousness must be considered, including long-felt but unsolved needs, failure of others, and commercial success. *In re Huang*, 100 F.3d 135, 139 (Fed. Cir. 1996) (citing *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966)). These indicia constitute independent evidence of nonobviousness that can be the most probative evidence in that regard “in avoiding the trap of hindsight when reviewing, what otherwise seems like, a combination of known elements.” *Leo Pharm. Prods., Ltd. v. Rea*, 726 F.3d 1346, 1358 (Fed. Cir. 2013).

The Board has repeatedly cautioned “that known evidence of secondary considerations should be addressed in the petition.” *See, e.g., Robert Bosch Tool Corp. v. SD3, LLC*, IPR2016-01751 (PTAB Mar. 22, 2017) Paper 15 at 23 (collecting prior Board decisions that deny institution based, at least in part, on a petitioner’s failure to address known evidence of secondary considerations). Despite Patent Owner providing evidence regarding the long-felt, unmet need for the invention in its district court complaint against Petitioner (EX-2011, at 24-28), Petitioner fails to consider such evidence in its petition; thus, institution should be denied.

As discussed above, a long-felt and persistent need existed to improve malware classification performance by dynamically calibrating feature vectors at the classification stage based on the content of the executable. (EX-2001 ¶ 108.) Prior art attempts were unable to improve classification performance by performing such dynamic calibration at the classification stage. Where others failed, the inventors of the '844 Patent succeeded.

Patent Owner currently accuses at least four industry participants, including Petitioner, of infringing the innovative technology claimed in the '844 patent, demonstrating the critical need and widespread industry reliance on these novel methods. (EX-2011, at 126-133.) Petitioner, for example, markets and provides the Intercept X platform, the components of which practice the systems and methods claimed in the '869 patent.

Accordingly, secondary objective indicia of non-obviousness confirm that Petitioner cannot show that it is more likely than not to succeed on any of the challenged independent claims.

IV. CONCLUSION

For the foregoing reasons, the Board should decline to institute a PGR of the '869 patent.

Patent Owner's Preliminary Response
to Petition for *Post-Grant Review*
Patent No. 11,409,869

Respectfully submitted,

KING & SPALDING LLP

BY: /Brian Eutermoser/

Brian Eutermoser
Registration No. 64,058

Date: September 6, 2023
1401 Lawrence Street, Suite 1900
Denver, CO 80202
Telephone: (720) 535-2312

CERTIFICATION OF WORD COUNT

The undersigned hereby certifies that the portions of the above-captioned PATENT OWNER'S PRELIMINARY RESPONSE specified in 37 C.F.R. § 42.24 includes 13,358 words and therefore complies with the 14,000-word limit set forth in 37 C.F.R. § 42.24. This word count was prepared using Microsoft Word 365 v.2022.

Date: September 6, 2023

Respectfully submitted,

KING & SPALDING LLP

BY: /Brian Eutermoser/
Brian Eutermoser
Registration No. 64,058

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** and all supporting **EXHIBITS 2001-2019** were served by filing the documents through the Patent Trial and Appeal Case Tracking System (P-TACTS) and courtesy copies were also served via electronic mail, in their entireties on the following:

Benjamin Hershkowitz
(Reg. No. 41,833)
GIBSON, DUNN & CRUTCHER LLP
200 Park Avenue
New York, NY 10166
Telephone: 212.351.2410
Facsimile: 212.351.6210
bhershkowitz@gibsondunn.com

Brian A. Rosenthal
(*pro hac vice motion requested*)
GIBSON, DUNN & CRUTCHER LLP
200 Park Avenue
New York, NY 10166-0193
Tel: (212) 351-2339
BRosenthal@gibsondunn.com

Katherine Q. Dominguez
(*pro hac vice motion requested*)
GIBSON, DUNN & CRUTCHER LLP
200 Park Avenue
New York, NY 10166-0193
Tel: (212) 351-2338
KDominguez@gibsondunn.com

Josh Krevitt
(*pro hac vice motion requested*)
GIBSON, DUNN & CRUTCHER LLP
1881 Page Mill Road
Palo Alto, CA 94304-1211
Tel: (212) 351-2490
JKrevitt@gibsondunn.com

Timothy P. Best
(*pro hac vice motion requested*)
GIBSON, DUNN & CRUTCHER LLP
333 South Grand Avenue
Los Angeles, California 90071-1504
Tel: (213) 229-7659
TBest@gibsondunn.com

Counsel for Petitioners
Sophos Ltd. and Sophos Inc.

Patent Owner's Preliminary Response
to Petition for *Post-Grant Review*
Patent No. 11,409,869

KING & SPALDING LLP

BY: /Brian Eutermoser/
Brian Eutermoser
Registration No. 64,058

Attorney for Patent Owner
Open Text Inc.

Date: September 6, 2023
1401 Lawrence Street, Suite 1900
Denver, CO 80202
Telephone: (720) 535-2312