

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SOPHOS LTD. and SOPHOS INC.,

Petitioners

v.

Webroot INC.,

Patent Owner

Case IPR2023-00528
Patent No. 10,599,844

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.	3
A.	Factor 1: The district court has not granted and will not grant a stay.....	6
B.	Factor 2: A district court trial is expected near the projected statutory deadline.	11
C.	Factor 3: The parties will have invested significant resources in the district court proceeding.....	13
D.	Factor 4: The issues and parties overlap significantly.	15
E.	Factor 5: Both parties are parties in the parallel litigation.....	18
F.	Factor 6: The petition does not present “compelling evidence of unpatentability.”	20
III.	THE BOARD SHOULD DENY INSTITUTION BECAUSE PETITIONER IS NOT REASONABLY LIKELY TO PREVAIL ON ANY CLAIM.....	21
A.	Background	22
1.	The state of the art.....	22
2.	The ’844 Patent	24
3.	Claim construction.	28
4.	The level of ordinary skill.....	30
B.	Ground 1: Ideses does not render Claims 1-20 obvious	30

Patent Owner’s Preliminary Response
to Petition for *Inter Partes* Review
Patent No. 10,599,844

1.	Overview of Ideses.....	30
2.	Ideses does not render claim 1 obvious because it fails to teach or suggest turning features on or off at the classification stage, based on the executable being classified.....	34
3.	Ideses fails to render claims 2-20 obvious.....	37
C.	Ground 2: Claims 1, 8-10, and 17 were not rendered obvious by Mori.	38
1.	Overview of Mori.....	39
2.	Mori fails to render claim 1 obvious because it does not teach or suggest turning features on or off at the classification stage based on the executable being classified.....	41
3.	Mori fails to render claims 8-10 and 17 obvious.....	47
D.	Ground 3: The combination of Mori and Ideses does not render claims 1-20 obvious.....	48
1.	Petitioner does not explain the proposed combination.	48
2.	Petitioner does not identify a sufficient motivation to combine Ideses and Mori.	49
3.	No combination of Ideses and Mori can render the independent claims obvious because neither reference teaches turning features on or off during the classification stage, based on the executable being classified.	51
IV.	THERE ARE OBJECTIVE INDICIA OF NONOBVIOUSNESS	51
V.	CONCLUSION.....	53

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 11 (PTAB Jun. 1, 2021)	<i>passim</i>
<i>Apple v. Seven Networks</i> , IPR2020-00255, Paper 13 (PTAB July 28, 2020)	14
<i>Commscope Techs. v. Dali Wireless, Inc.</i> , IPR 2022-01242, Paper 23 (PTAB Feb. 27, 2023)	21
<i>Google LLC v. EcoFactor, Inc.</i> , IPR2021-00454, Paper 9 (PTAB July 22, 2021)	17
<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966)	51-52
<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)	52
<i>Luxshare Precision Industry v. Amphenol Corp.</i> , IPR2022-00132, Paper 10 (PTAB May 3, 2022)	10
<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>Nintendo Co. v. Gamevice, Inc.</i> , IPR2020-01197, Paper 13 (PTAB Jan. 12, 2021)	10

Patent Owner’s Preliminary Response
to Petition for *Inter Partes* Review
Patent No. 10,599,844

<i>Nokia of Am. Corp. v. Godo Kaisha IP Bridge</i> , IPR2022-00755, Paper 9 (PTAB Nov. 2, 2022)	18
<i>NXP USA, Inc. v. Impinj, Inc.</i> , PGR2022-00005, Paper 20 (PTAB Aug. 25, 2022)	12, 13
<i>Robert Bosch Tool Corp. v. SD3, LLC</i> , IPR2016-01751 (PTAB Mar. 22, 2017)	52
<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>Sotera Wireless, Inc. v Masimo Corp.</i> , IPR2020-01019, Paper 12 (PTAB Dec. 1, 2020)	16, 17
<i>Twinstrand Biosciences, Inc. v. Guardant Health, Inc.</i> , IPR2022-01115, Paper 14 (PTAB Dec. 30, 2022)	38, 48
<i>Vizio, Inc. v. Maxell, Ltd.</i> , IPR2022-01459, Paper 7 (PTAB Apr. 11, 2023)	18
<i>Webroot v. AO Kaspersky Lab</i> , No. 6:22-cv-00243-ADA-DTG (WDTX)	1
<i>Webroot v. Crowdstrike Inc.</i> , No. 6:22-cv-00241-ADA-DTG (WDTX)	1
<i>Webroot v. Forcepoint LLC</i> , No. 6:22-cv-00342-ADA-DTG (WDTX)	1-2
<i>Webroot v. Sophos Ltd.</i> , No. 6:22-cv-00240-ADA-DTG (WDTX)	1

Webroot v. Trend Micro Inc.,
No. 6:22-cv-00239-ADA-DTG (WDTX).....1

Statutes

35 U.S.C. § 314(a) 2, 3, 20

Regulations

37 C.F.R. § 42.100(b) 28

Other References

Director’s Memo, *Interim Procedure for Discretionary Denials in AIA
Post-Grant Proceedings with Parallel District Court Litigation* (June
21, 2022) 3, 4, 12, 13, 20
H.R. Rep. No. 112–98, 2011 U.S.C.C.A.N 1
Matthews Annotated Patent Digest (Third Edition) 50

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, Order Granting Third Amended Scheduling Order, Dkt. 142 (W.D. Tex., Dec. 27, 2022)
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, Claim Construction Order, Dkt. 236 (W.D. Tex. Mar. 16, 2023)
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	<i>Webroot, Inc., Open Text Inc., v. AO Kaspersky Lab et al.</i> , Case No. 6:22-cv-00243-ADA-DTG, Order Granting Third Amended Scheduling Order, Dkt. 142 (W.D. Tex., Dec. 27, 2022)

Patent Owner’s Preliminary Response
to Petition for *Inter Partes* Review
Patent No. 10,599,844

Exhibit	Description
2010	United States Courts, Statistics & Reports, Federal Court Management Statistics—Comparison Within Circuit—During the 12-Month Period Ending June 30, 2022, “Comparison of Districts Within the First Circuit — 12-Month Period Ending June 30, 2022,” <i>available</i> https://www.uscourts.gov/sites/default/files/fcms_na_distcomparison0630.2022_0.pdf
2011	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-00240-ADA, Complaint for Patent Infringement, Dkt. 1 (W.D. Tex., Mar. 4, 2022)
2012	<i>Webroot, Inc. and Open Text Inc., v. ForcePoint LLC</i> , Case No. 6:22-cv-00342-ADA-DTG, Defendant ForcePoint LLC’s Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022) (excerpted)
2013	<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 (W.D. Tex., Sept. 13, 2022) (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., Sept. 13, 2022) (excerpted)
2015	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab</i> , Case No. 6:22-cv-00243-ADA-DTG, Defendant’s Preliminary Invalidity Contentions Appendix A (W.D. Tex., Sept. 14, 2022) (excerpted)
2016	<i>Webroot, Inc. and Open Text Inc., v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 6:22-cv-00241-ADA-DTG, Defendant’s Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022) (excerpted)

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 *inter partes* review of U.S. Patent No. 10,599,844 (“the ’844 patent”).

I. INTRODUCTION

The Board should deny institution for two reasons: (1) discretionary denial under *Fintiv* is appropriate because the ’844 patent is part of an exceptionally complex parallel litigation that is scheduled for trial within days of a final written decision, and (2) the petition is not reasonably likely to succeed because it relies on art that fails to disclose key claim limitations.

First, the Board should deny institution under *Fintiv* because institution would undermine the congressional purpose of IPRs to “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. The validity of the ’844 patent is currently being litigated in *five* consolidated district court cases in the Western District of Texas. *See Webroot v. AO Kaspersky Lab*, No. 6:22-cv-00243-ADA-DTG (WDTX) (lead case) (“Kaspersky”); *Webroot v. Trend Micro Inc.*, No. 6:22-cv-00239-ADA-DTG (WDTX); *Webroot v. Sophos Ltd.*, No. 6:22-cv-00240-ADA-DTG (WDTX); *Webroot v. CrowdStrike Inc.*, No. 6:22-cv-00241-ADA-DTG (WDTX); *Webroot v.*

Forcepoint LLC, No. 6:22-cv-00342-ADA-DTG (WDTX). Those five cases are scheduled for trial starting Monday, August 19, 2024 (EX-2003 at 9)—within three calendar days (and one court day) of the statutory deadline for a final written decision on this petition (assuming the petition were instituted three months after this POPR).

The five cases will not be stayed in response to institution because they involve *39 patents* and four additional parties, each of which could continue litigating the validity of the '844 patent in the district court based on the same grounds and references raised in this petition. The parallel cases have now advanced past claim construction (EX-2007), and the parties have invested significant resources in that litigation. Consequently, the only possible outcome of institution would be to needlessly increase litigation costs by forcing the parties to litigate validity before ***both*** the Board and the district court. The Board should decline to institute on this ground alone.

Second, the Board should deny institution under 35 U.S.C. § 314(a) because the petition fails to “show[] that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” § 314(a). Most saliently, none of the cited art (alone or in combination) discloses, suggests, or renders obvious the claimed idea of a malware classification technique

that turns features on or off *at the classification stage, based on the executable being classified*. Instead, the cited art discloses *training* of malware classifiers in the standard manner—i.e., identifying and selecting features *at the training stage*, and then, at the subsequent classification stage, extracting and evaluating *all* those preselected features, irrespective of the executable being classified. Because none of the cited art suggests turning features on or off at the *classification stage based on the executable being classified*, the petition is not reasonably likely to succeed and must therefore be denied.

II. THE BOARD SHOULD EXERCISE ITS DISCRETION TO DENY INSTITUTION.

In view of the parallel litigations, in which a district court will adjudicate the same grounds raised in this petition, and on the same timeline as the Board, the Board should exercise its discretionary denial authority under 35 U.S.C. § 314(a).

“Congress designed the America Invents Act (AIA) post-grant proceedings to ... limit unnecessary and counterproductive 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). Consequently, Director Vidal has instructed the Board not to institute IPR trials that, in view of parallel litigation, would “increase, rather than limit, litigation costs.” *Id.*

Here, the proceedings Petitioner seeks would do just that. 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, for the next year, the Board and the district court will labor to decide the very same issues, and even if the Board does finish first by a few days, the year-long race between the two adjudicatory bodies will nevertheless “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 1. This is exactly the situation that Director Vidal has instructed the Board 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 Jun. 1, 2021).

Confirming this conclusion, every one of the *Fintiv* factors weighs for discretionary denial:

- (1) The likelihood of a stay of the co-pending district court litigation is effectively nil because Petitioner has petitioned for review of only seven of the thirteen patents asserted against it and because the '844 patent is currently asserted against four other defendants in a now-consolidated

case, where the claims have already been construed and discovery is well underway.

- (2) Trial in the district court is expected within three calendar days (and one court day) of a final written decision.
- (3) The parties and district court have already invested heavily in carrying that 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 Petitioner's co-defendants, who have raised the same grounds before the district court and will maintain those grounds irrespective of the Board's institution decision.
- (5) Petitioner will get its chance to make its arguments before the district court because Petitioner is a defendant in the parallel litigation.
- (6) And, as shown below, the grounds presented in the petition are weak.

Thus, every *Fintiv* factor here weighs for discretionary denial.

The goals of the AIA and the general welfare of the patent system would be best advanced here by denying institution and allowing the district court to resolve the dispute presented in this petition.

A. Factor 1: The district court has not granted and will not grant a stay.

Fintiv factor 1, which considers the likelihood of a stay in the parallel district court litigation, weighs strongly for discretionary denial because no stay has been requested or ordered, and the circumstances and timing of the district court case demonstrate that a stay is out of the question.

The first *Fintiv* factor asks “whether the court granted a stay or evidence exists that one may be granted if a proceeding is instituted.” *Fintiv* at 6. Here, the undisputed answer to both questions is **no**: the court has not “granted a stay,” and Petitioner has identified no “evidence [] that one may be granted if a proceeding is instituted.” *Id.*

On the other side of the scale, specific evidence indicates that a stay is highly unlikely, including (1) the complexity of the parallel litigation, such that a stay will not simplify the issues before the district court, and (2) the timing of the case, with trial contemporaneous with a final written decision. The district court has been clear that it will refuse a stay in either set of circumstances.

First, a stay is unlikely because a stay would not simplify the issues before the court. The district court’s record demonstrates that, when determining a stay, it

considers issue simplification “the most important factor.”¹ But here, no simplification is on offer. The validity of the '844 patent is currently only one of many issues being litigated in a consolidated action involving 39 patents and *five* defendants. The case against Petitioner alone involves twenty patents—thirteen asserted by Patent Owner and seven by Petitioner, with all but one unrelated to the '844 patent.

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

¹ 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).

Patent Owner's Preliminary Response
to Petition for *Inter Partes* Review
Patent No. 10,599,844

Patent	K	CS	FP	S	TM	IPR
8,726,389	X	X	X	X	X	IPR2023-00556-CS; IPR2023-00633-S
10,599,844	X	X	X	X	X	IPR2023-00528-S; IPR2023-00662-TM
8,418,250	X	X		X	X	IPR2023-00289-CS; IPR2023-00655-TM; IPR2023-00699-S
9,413,721			X	X		IPR2023-00491-S
9,578,045	X	X		X	X	IPR2023-00124-CS; IPR2023-00677-TM
10,257,224	X	X		X	X	IPR2023-00126-CS; IPR2023-00657-TM
10,284,591	X	X		X	X	IPR2022-01522-CS; IPR2023-00692-TM; IPR2023-00895-K
8,856,505	X	X		X	X	
8,181,244		X		X	X	
8,201,243	X	X	X	X	X	IPR2023-00731-S
8,719,932	X	X	X	X	X	IPR2023-00732-S
11,409,869	X	X	X	X	X	PGR2023-00031-S
8,763,123		X		X		
8,438,386			X			IPR2023-00784-FP
10,025,928			X			IPR2023-00786-FP

Table 1. Patents Asserted by Patent Owner in Parallel Cases
("K" – Kaspersky, "CS" – CrowdStrike, "FP" – Forcepoint, "S" – Sophos,
"TM" Trend Micro; Green Shading – no IPR petition filed)

As it must, Petitioner concedes that “this multi-defendant consolidated case is exceptionally complicated” and “especially complex.” (Petition 67-68.) Indeed, it cites the district court’s recognition that “this case is ‘about as complicated as you’re going to get with patent cases.’” (*Id.*)

In sum, that the parallel litigation involves so many claims and issues unrelated to the validity of the ’844 patent means that the district court will not stay the consolidated proceedings in response to institution.

Second, the district court has been clear in public comments that it will not order a stay in cases where, as here, the district court can resolve the case as quickly as the Board can. For example, Judge Albright has opined that:

there’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 28.3 months, meaning that trial is expected in this case in mid-July 2024, before a final written decision (expected August 16, 2024). That Judge Albright has made public comments indicating that his practice is to deny

a stay in precisely the circumstances presented here suggests that he would not stay this case in response to institution of this petition.

Petitioner argues that Factor 1 can only favor institution or be neutral because the Board will not attempt to predict how district courts will proceed. (Petition 65.) But that is false: 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). That is particularly so where (as here), the “record does not indicate that Petitioner has requested a stay,” but the circumstances do show that “a stay ... is unlikely.” *Nintendo Co. v. Gamevice, Inc.*, IPR2020-01197, Paper 13, at 11-12 (PTAB Jan. 12, 2021).

For example, in *Samsung Elecs. Co. v. Clear Imaging Research LLC*, the Board held that 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 of the patentability issues...before proceeding with the parallel litigation.” IPR2020-01399, Paper 13 at 11-12 (PTAB Feb. 3, 2021). While the district court in that case had denied an earlier motion to stay *without* prejudice, the Board held that the court’s statements regarding prejudicial delay and the expenditure of resources showed that a stay was unlikely. *Id.* at 10-13. Thus, although the petitioner could

still seek a stay, the Board found that factor 1 “weigh[ed] slightly in favor of exercising [] discretion to deny institution.” *Id.* at 12.

In *Motorola Mobility LLC v. Ironworks Patents, LLC*, the Board similarly found that factor 1 weighed in favor of denying institution, even though the petitioner never moved for a stay. IPR2021-00421, Paper 11 at 7-8 (PTAB July 22, 2021). The Board reached that conclusion because the district court’s scheduling guidelines allowed motions for a stay pending IPR only under “exceptional circumstances,” thus indicating that a stay was unlikely. *Id.* at 7-8.

When one considers the full circumstances of this case, the only plausible conclusion regarding a stay is that the district court will not grant one. Because institution will not result in a stay, Factor 1 weighs for discretionary denial. *Samsung*, IPR2020-01399, Paper 13 at 11-12; *Motorola*, IPR 2021-00421, Paper 11 at 7-8.

B. Factor 2: A district court trial is expected near the projected statutory deadline.

Factor 2, which considers the implications of the district court and IPR schedules, also weighs for denial (or is neutral at best). The factor requires more than simple prognostication over whether the district court or the PTAB will nose over the adjudicatory finish line first. As noted above, *Fintiv* instructs the Board to take “a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review,” *Fintiv* at 6, and Director Vidal has instructed the

Board to deny petitions whose institution would “increase, rather than limit, litigation costs.” Guidance at 1.

That is precisely what institution here would do because the IPR trial would overlap almost entirely with the parallel district court litigation. WDTX’s median time-to-trial as of June 2022 is 28.3 months (EX-2010 at 5), indicating a trial by mid-July 2024—a date consistent with the currently scheduled trial date of August 19, 2024. (EX-2009 at 9.) Should the Board institute, its final written decision deadline would be August 16, 2024, a month *after* WDTX’s median time-to-trial and three calendar days (one court day) before the scheduled trial date. In other words, the IPR proceedings will overlap almost entirely with the parallel litigation.

Because no stay will be granted and the district court schedule overlaps almost entirely with the IPR schedule, the Board and district court will proceed in parallel for a year, resulting in precisely the “duplicative,” “unnecessary[,] and counterproductive litigation costs” that *Fintiv* and the AIA were designed to avoid. *Fintiv* at 10; Guidance at 1. That is so, regardless of whether the Board reaches a final adjudication just before or just after the district court. In either case, the Board should consider “the duplicative efforts that will occur during the [] duration of the inter partes review.” *NXP USA, Inc. v. Impinj, Inc.*, PGR2022-00005, Paper 20 at 7 (PTAB Aug. 25, 2022).

Petitioner speculates that the district court may ultimately postpone the scheduled trial date further (Petition 65-68), but that speculation carries little weight because the parties do not rely only on the scheduled trial date, but on median time-to-trial statistics, as the Director has recommended. Guidance at 8-9. Moreover, Petitioner's assertion that delay is likely due to case complexity is unsupported by evidence and cannot, by itself, weigh against exercising discretion. *See NXP*, PGR2022-00005, Paper 20 at 8.

Because the district court will not stay the litigation, and the litigation schedule would overlap the IPR trial schedule almost entirely, the second *Fintiv* factor also favors discretionary denial.

C. Factor 3: The parties will have invested significant resources in the district court proceeding.

Factor 3, which considers the resources invested in the parallel district court proceeding, weighs strongly for denying institution.

The parties have invested significant resources in the parallel litigation over the last 14 months. For example, the parties long ago exchanged infringement and invalidity contentions (in July 2022 and September 2022 respectively), and fact discovery is underway. Moreover, the district court issued its claim construction order two months ago, on March 16, 2023. (EX-2007.) Under *Fintiv*, "claim construction orders may indicate that the court and parties have invested sufficient time in the

parallel proceeding to favor denial.” *Fintiv* at 10. And while *Fintiv* directs consideration of resource investment at the time of institution, the overlapping schedules here counsel the Board to further consider the redundant resources that *will be* wasted should IPR be instituted. *Id.* at 6 (requiring “a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review”).

Petitioner's delay in filing this petition further weighs for denial. Petitioner waited eleven months after receiving Patent Owner's March 4, 2022 complaint, seven months after receiving Patent Owner's infringement contentions, and five months after serving its invalidity contentions before filing its petition. (EX-2011 at 1-2). Petitioner offers no explanation for its delay, which weighs for denial. *Fintiv* at 11-12.

Petitioner argues that this investment of time and resources by the parties and the district court should be “counterbalance[d]” by Petitioner's investment in the petition. (Petition 68.) In support of this argument, Petitioner cites *Apple v. Seven Networks*, IPR2020-00255, Paper 13 at 12-15 (PTAB July 28, 2020). But, in *Apple*, the Board merely noted that the petitioner had made this argument (because the petitioner had filed 21 petitions) but the Board did not give the argument any weight. The Board should do the same here because every petitioner spends resources drafting a petition, and Petitioner here does not demonstrate why its efforts should

warrant any special consideration. To the contrary, as compared to the significant resources the parties have already invested in the district court proceeding (including claim construction briefing and a hearing, infringement contentions, invalidity contentions, and ongoing discovery), Petitioner's investment in the petition is minor.

D. 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 weighs strongly for discretionary 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* 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.*

Here, the petition identifies substantially the same prior art arguments that Petitioner and multiple co-defendants have put before the district court. (Those other defendants are relying on the same prior art references cited in the petition, and none has offered any type of stipulation. (EX-2012 1, 59, 61; EX-2013 1, 18; EX-2014 1, 52; EX-2015 1, 6; EX-2016 1, 14, 15). This fact favors denial.

In evaluating this fourth factor, however, the Board has considered whether a petitioner's stipulation not to pursue certain defenses in the district court may "mitigate[]" any concerns of duplicative efforts between the district court and the Board, as well as concerns of potentially conflicting decisions." *Sotera Wireless, Inc. v. Masimo Corp.*, IPR2020-01019, Paper 12 at 19 (PTAB Dec. 1, 2020) (precedential). The Board has held that, at least where the petitioner is the only defendant in the district court, a stipulation not to pursue at the district court "any ground raised or that could have been reasonably raised" in the petition is effective, *id.*, whereas a limited stipulation not to pursue only the same grounds weighs only "*marginally* in favor of not exercising discretion to deny institution." *Sand Revolution II, LLC v. Cont'l Intermodal Group-Trucking LLC*, IPR2019-01393, Paper 24 at 11-12 (PTAB June 16, 2020) (emphasis added).

Here, Petitioner has offered only the limited stipulation that it would not use in the district court any prior art reference relied upon in the petition (Petition 69), and Petitioner's co-defendants have offered no stipulation. Petitioner argues that Factor 4 favors institution because Petitioner's "stipulation is significantly broader than the stipulation in *Sand Revolution*" (which related to only the same grounds rather than the same art). *Id.*

But Petitioner's stipulation (to not use the same prior art references) is certainly not "significantly broader" than a "marginal" *Sand Revolution* stipulation (to not use the same grounds). *Sand Revolution II* at 12 n.5. Even if technically broader, Petitioner's stipulation does no more than a *Sand Revolution* stipulation to "ensure that [the] IPR functions as a true alternative to litigation." *Sand Revolution II* at 12 n.5. To the contrary, the stipulation merely *compounds* litigation costs by allowing Petitioner to pursue separate invalidity theories in parallel at two different tribunals. Thus, the best that Petitioner's stipulation could do is "weigh[] only marginally in favor of [institution]." *Id.* at 11-12. The stipulation cannot carry any more weight.

And under the circumstances here, the stipulation carries *no weight* because it does not bar the *other defendants* in the parallel litigation from litigating the very same invalidity challenges before the district court. Those other defendants are relying on the same prior art references cited in the petition, and none has offered any type of stipulation. (EX-2012 1, 59, 61; EX-2013 1, 18; EX-2014 1, 52; EX-2015 1, 6; EX-2016 1, 14, 15). Consequently, Petitioner's stipulation does not "mitigate[] any concerns of duplicative efforts between the district court and the Board," or "of potentially conflicting decisions." *Sotera* at 19; *see Google LLC v. EcoFactor, Inc.*, IPR2021-00454, Paper 9 at 12-14 (PTAB July 22, 2021) (denying institution based on *Fintiv*, even where co-defendants in the parallel litigation provided matching

Sands Revolution stipulations). For these reasons, Factor 4 weighs in favor of discretionary denial.

E. Factor 5: Both parties are parties in the parallel litigation.

Factor 5—which considers overlap of the parties between the petition and the parallel litigation—favors denial because (1) Petitioner is a defendant in the parallel litigation and will therefore get the opportunity to litigate the petitioned grounds there, and (2) the issues raised in the petition are the same ones *also* raised in the parallel litigation by other defendants. Both factors favor discretionary denial.

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* 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 weighs yet further for discretionary 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.” *Id.* 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 fails to offer such an explanation here. That other defendants are currently litigating the same issues raised in the petition raises concerns of duplicative work by adjudicators, and any public policy concerns regarding the validity of patents 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.

Petitioner argues that Factor 5 weighs in favor of institution because “Patent Owner has a demonstrated history of asserting claims against multiple defendants, and the Board has an opportunity to invalidate some of those claims here.” (Petition 70.) Petitioner cites neither legal precedent establishing such a rule nor logical

rationale supporting it. Neither exists. The Board has held that, when multiple parties are litigating the petitioned issues before a district court, this weighs for *denial*, not for institution. And this is sensible, given that adjudication by the district court is no less effective a remedy than adjudication by the Board. Factor five therefore favors denial.

F. Factor 6: The petition does not present “compelling evidence of unpatentability.”

Factor 6—which considers the merits of the Petition—also weighs against institution because, as outlined in the following section, the petition fails to show a “reasonable likelihood” that Petitioner would prevail on any challenged claim. *See* 35 U.S.C. § 314(a). For example, none of the cited art teaches, suggests, or renders obvious the idea of a machine-learning malware classifier that turns features on or off at the classification stage (let alone based on the values of static data points extracted from the executable being classified at that stage). Because the petition fails to show a reasonable likelihood of success on the merits, the petition may not be instituted.

Even if the petition did pass the threshold of “reasonable likelihood” (it does 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, 3 (PTAB Feb. 27, 2023). That higher standard requires determining that the Petitioner’s chances of succeeding on the merits are “highly likely.” *Id.* at 4. But just the opposite is true here. As shown below, Petitioner’s grounds are exceptionally weak.

* * *

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 REASONABLY LIKELY TO PREVAIL ON ANY CLAIM.

The Board should deny institution because the petition fails to show a reasonable likelihood that Petitioner can prevail on any claim. The petition presents three grounds:

Ground 1	Section 103(a)	Obvious over Ideses	Claims 1-20
Ground 2	Section 103(a)	Obvious Mori	Claims 1, 8-10, 17
Ground 3	Section 103(a)	Obvious over the combination of Ideses and Mori	Claims 1-20

The central problem with these grounds is that none of the cited art (alone or in combination) discloses or suggests a classifier that selectively turns features on or off at the classification stage, based on the executable being classified, as claimed.²

A. Background

1. The state of the art

Every day, thousands of new executable files (“executables”) are created and distributed across computer networks. (EX-1002 1:7-8.) 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.* 1:8-15.)

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.* 2:15-17.) 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

² 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 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.

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.* 1:20-40.) 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 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, at ¶¶ 21-22.)

An 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 ¶ 23.)

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 ¶ 24.)

2. The '844 Patent

The '844 patent improves upon the prior art by introducing an executable classification technique that performs feature selection at the *classification stage, based*

on 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 7:57-8:22; 12:54-13:4.) 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.* 7:57-8:22.)

To decide which features should be turned on or off, the patent teaches analyzing “static data points” in the executable. 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 5:62-6:58, 8:53-56.)

The specification provides many examples of static data points (*see id.* Table 1.1, 8:53-56), and a person of ordinary skill would recognize that the provided examples correspond to fields defined by the “Portable Executable” (or “PE”) file format. (EX-2001 ¶¶ 29-30; EX-1001 3:35-46 (discussing the extraction of static data points from PE files as the preferred but not exclusive embodiment), *id.* 4:22-24

(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 8:53.)

The claimed system can turn features on or off *during the classification phase* based on the values of the static data points extracted from the executable being classified. (*Id.* 5:62-6:58, 8:53-56, 7:63-66, 8:63-9:7 (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.* 8:49-9:40.)

Claim 1 recites a computer-implemented method corresponding to the classification stage of the invented technique:

1. A computer-implemented method comprising:

extracting a plurality of static data points from an executable file without decrypting or unpacking the executable file, wherein the plurality of static data points represent predefined character strings in the executable file;

generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files,

wherein the collection of data comprises at least a portion of the plurality of static data points, and

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 extracting static data points from the executable being classified. As described above, the static data points represent predefined character strings in the executable (*e.g.*, a “File Size” field of a PE file). The method then calls for using a classifier that was previously “trained” (past tense) on “known malicious,” “known benign,” and “known unwanted” executables that included at least some of the same static data points. The classifier generates a feature vector from the static data points it extracted from the executable, and it turns certain features on or off if the values of those static data points are within certain ranges. 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.

Claim 17 recites a system that performs a variation of the method recited in claim 1: the system of claim 17 extracts predefined character strings (rather than static data points) and uses those to create a feature vector while turning certain features on or off based on the values of those predefined character strings.

3. 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 an interlocutory Claim Construction Order rejecting various proposals advanced by Petitioner and other defendants. (EX-2007 17-20.) The Board's regulations require that the Board consider those constructions. 37 C.F.R. § 42.100(b).

- **“executable file.”** The district court held that the phrase “executable file” carries its plain and ordinary meaning. (EX-2007 17.) The court rejected a proposal that would have required “executable files” to be “encrypted or packed.” (*Id.*)
- **“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. (*Id.* 18.)

- **“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.” (*Id.*) The district court rejected Sophos’s argument that a value “within a predetermined range” must be “between a predefined lower and upper limit.” (*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.*) The court rejected Sophos’s argument 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.* 15.)

4. 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 11.)

B. Ground 1: Ideses does not render Claims 1-20 obvious

Petitioner fails to establish a reasonable likelihood that Ideses renders independent claims 1, 10, and 17 obvious. Petitioner likewise fails to establish that Ideses renders any of the dependent claims obvious.

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].) That idea differs markedly from the method disclosed in the '844 patent, which is that of selecting features at the *classification* stage, *based static data points extracted from the executable being classified*. Because Ideses fails to disclose these limitations, it does not render the claimed invention obvious.

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. Both are reproduced below for the Board’s convenience.

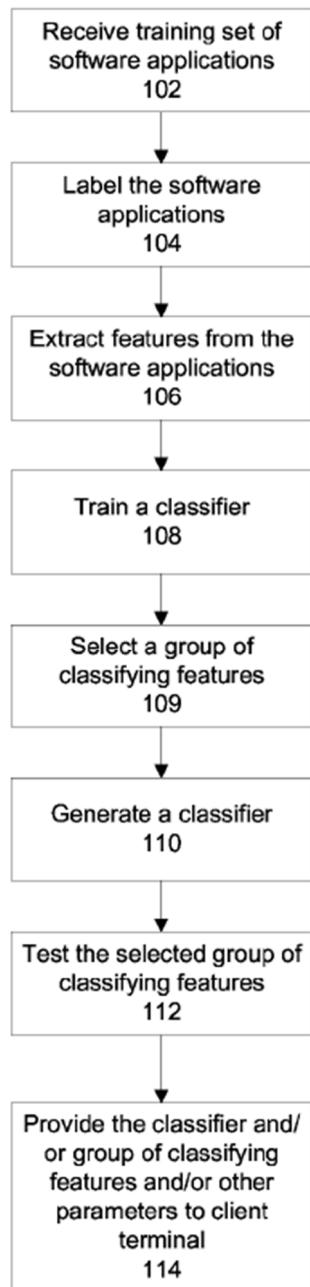


Fig. 1 (Training)

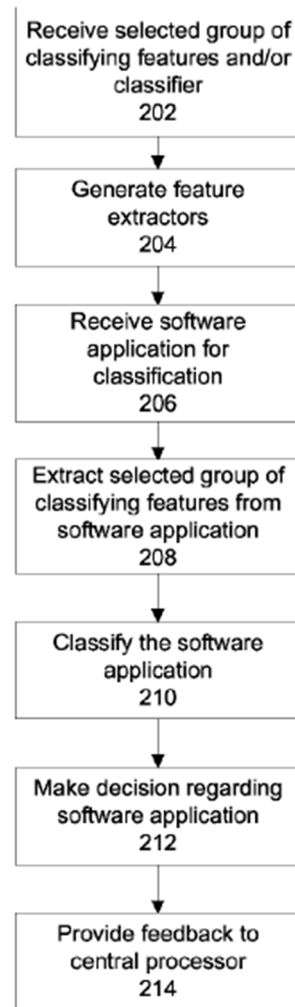


Fig. 2 (Classification)

(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 does not render claim 1 obvious because it fails to 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 claim 1 of the '844 patent is starkly different 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, and there is no feature selection at the classification stage. In contrast, the idea claimed in the '844 patent

is to turn features on or off at the *classification* stage, based on the values of data points extracted from the executable being classified at that stage. Ideses does not teach or suggest that approach.

Recall that claim 1 recites the steps of the *classification* stage:

- (1) “extracting ... static data points from an executable,”
- (2) “generating a feature vector from the ... static data points **using a classifier trained to classify (past tense)** the ... static data points,” “wherein one or more features of the feature vector are ... turned on or off based on ... [the] extracted static data points,” and
- (3) “evaluating the feature vector ... to determine whether the executable ... is harmful” (*i.e.*, classifying the executable).

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, challenged claim 1 recites the idea of turning features on or off *at the classification stage*, after the classifier has been trained to classify the static data points 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 ¶ 26.)

Ideses fails to disclose turning features on or off at the classification stage based on the executable being classified. As discussed above, challenged claim 1 recites the steps of the classification stage, wherein the system turns features on or off based on static data points extracted from the executable being classified at that classification stage. 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 30.) 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 ¶¶ 64-67.)

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 claim 1 of the '844 patent. (*See* EX-2001 ¶¶ 67-68.)

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 30.) 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 in claim 1 of the '844 patent.

3. Ideses fails to render claims 2-20 obvious.

Regarding independent claims 10 and 17, Petitioner presents no independent argument beyond those it presents for claim 1. (*See* Petition 42, 44-46.) Accordingly,

Petitioner's arguments regarding independent claims 10 and 17 fail for the same reasons as those regarding claim 1.

Regarding the dependent claims, because Petitioner has failed to establish a reasonable likelihood of success 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).

Accordingly, Petitioner has failed to show a reasonable likelihood of prevailing on Ground 1, with respect to any of the challenged claims.

C. Ground 2: Claims 1, 8-10, and 17 were not rendered obvious by Mori.

Petitioner argues in Ground 2 that Mori renders claims 1, 8-10, and 17 obvious. That is wrong because Mori has similar failings to Ideses: it teaches feature selection *at the training stage*, not at the classification stage based on the executable being classified.

1. Overview of Mori³

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. The feature vectors each represent *M* features,

³ The following analysis is based on the translation of Mori Petitioner supplied. Patent Owner does not concede it is accurate and reserves the right to obtain its own certified translation.

where 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 56.) 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.

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 claim 1 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 claim 1 obvious because 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*. Specifically, claim 1 recites: (1) “extracting ... static data points from an executable,” (2) “generating a feature vector from the ... static data points using a classifier **trained to classify** (past tense) the ... static data points,” “wherein one or more features of the feature vector are ... turned on or off based on ... [the] extracted static data points,” and (3) “evaluating the feature vector ... to determine whether the executable ... is harmful” (*i.e.*, classifying the executable). Thus, the '844 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.* 7:62-66.)

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.) Turning features on or off at the classification stage therefore improves upon prior art processes, where the classifier had to determine values for every feature predetermined in the training stage and consider those values in the classification stage. (*Id.*)

Mori's system does not do this. 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: (1) the classifier setting the value of a feature to 0 when the readable word corresponding to that feature is absent (Petition 54-55), and (2) the classifier purportedly omitting 0-value features from the feature vector (Petition 56). Neither teaching suggests turning features “on or off” at the classification stage based on the executable being classified.

First, Petitioner argues that Mori's classifier turns a feature “off” when it sets the value of that feature to 0, which would happen when the readable word corresponding to that feature does not appear in the executable. (Petition 54-55.) 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 process of classifying the executable. (EX-2001 ¶ 79.) The value of 0 simply indicates that the readable word corresponding to that feature does not appear in the executable. (EX-2001 ¶ 80.) Mori's classifier considers that information when it classifies the executable no less than it considers the information conveyed by other features whose value is not 0. (*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.

Second, Petitioner argues that Mori turns a feature “off” by *excluding* the feature from the feature vector, rather than by setting its value to 0. (Petition 56.) Petitioner seems to base this argument on the example feature vectors partially shown on each row of Mori Table 5, where each feature vector appears as though it may include entries only for the readable words found in the executable (*i.e.*, features with non-zero values). (Petition 56.) Petitioner argues that the features included in the exemplary feature vectors are thus turned “on” whereas features not included (*i.e.*, features with 0-values) are turned “off.” (Petition 56 (“Thus, 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”).))

Petitioner’s argument is contradicted by Mori’s full disclosure. Mori expressly discloses that features with 0-values *are* included in the extracted feature vectors. (EX-1006 [0030].) For example, 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 ¶¶ 81-82.)

Petitioner's reading is further belied by the fact that 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 is correct 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.*, 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 further demonstrates that Mori does not disclose a classifier that can turn features on or off at the classification stage, as challenged claim 1 recites.

Mori's disclosure in Table 5 (of feature vectors that appear to omit features corresponding to readable words not in the executable) 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 would 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

¶ 30.) 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 on which the classifier was trained to accept. (EX-2001 ¶ 83-84.) Thus, 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. Consequently, 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 claims 8-10 and 17 obvious.

Regarding independent claims 10 and 17, Petitioner presents no independent argument beyond those it presents for claim 1. (*See* Petition 58-61.) Accordingly,

Petitioner's arguments regarding independent claims 10 and 17 fail for the same reasons as those regarding claim 1.

Regarding the dependent claims, because Petitioner has failed to establish a reasonable likelihood of success on any of the *independent* claims, it cannot meet its burden for the challenged dependent claims. *Twinstrand Biosciences*, IPR2022-01115, Paper 14, 19. Accordingly, Petitioner has failed to show a reasonable likelihood of prevailing on Ground 2 with respect to any of the challenged claims.

D. Ground 3: The combination of Mori and Ideses does not render claims 1-20 obvious.

In its final ground, Petitioner argues that the combination of Mori and Ideses renders all 20 claims of the '844 patent obvious. There are at least three problems with this argument: (1) Petitioner does not explain what the combination is, (2) Petitioner does not explain why a skilled artisan would have been motivated to make that combination, and (3) in any event, no combination of Ideses and Mori can render the claimed invention obvious because, as shown above, neither reference teaches, suggests, or renders obvious the idea of turning features on or off during the classification stage.

1. Petitioner does not explain the proposed combination.

The first problem with Ground 3 is that Petitioner does not explain the hypothetical combination of Ideses and Mori. Petitioner uses Ideses as the primary

reference. (Petition 61 (if “Ideses does not sufficiently teach or suggest any element of any of the challenged independent claims, Mori discloses these features”).) And Petitioner argues that it would have been obvious to modify Ideses with “Mori’s ‘readable strings’ malware detection technique.” (*Id.*) But Petitioner never explains *how* the combined system would operate—*i.e.*, what it means “to have combined Mori’s ‘readable strings’ malware analysis technique with Ideses’ malware detection method.” (*Id.*) The Petition thus leaves Patent Owner and the Board guessing at the proposed combination and thus makes evaluating Petitioner’s argument difficult.

Because the Petition fails to articulate the proposed combination, it fails to demonstrate that the combination renders the challenged claims obvious.

2. Petitioner does not identify a sufficient motivation to combine Ideses and Mori.

The second problem with Ground 3 is that Petitioner fails to identify a sufficient reason to combine the references resulting in a petition that reeks of improper hindsight analysis.

The only motivation to combine articulated in the petition is that a skilled artisan would have been motivated to combine Mori with Ideses “to enable Ideses’ method to be applied to encrypted and/or packed executable files without decrypting and unpacking those files.” (Petition 61-62; *see also id.* 64 (“A POSITA would have

been motivated to use Mori's examples of predefined 'readable strings ...' for the same reasons described above.'').)

But that argument conflicts with Petitioner's earlier argument that Ideses already teaches that "feature extraction would be conducted ***without decrypting*** or unpacking the executable file." (Petition 21 (emphasis added).) Indeed, Petitioner's expert opined that, in Ideses, "features are extracted '***without decrypting*** or unpacking the executable file.'" (Ex. 1004 ¶ 70 (emphasis added).) And on that basis, Petitioner has argued in Ground 1 that "[i]t would have been obvious to a POSITA reading Ideses to perform 'off-line' extraction of the 'static meta data' ***without attempting to first decrypt*** or unpack the executable file." (Petition 22 (emphasis added).)

Petitioner's position that Ideses already teaches extracting features "without decrypting or unpacking the executable files" (Petition 21) belies Petitioner's later position that a skilled artisan would have been motivated to combine Ideses and Mori "to enable Ideses' method to be applied to encrypted and/or packed executable files without decrypting and unpacking those files." (*Id.* 61-62.) "Generally, a motivation to modify or combine will not be found if the resulting feature will be merely redundant to what the prior art device already had." *Matthews Annotated Patent Digest (Third Edition)*, § 18:53.50 (collecting cases).

Because the Petition fails to articulate a sufficient motivation to combine Ideses and Mori, it fails to demonstrate that their combination renders the challenged claims obvious.

3. No combination of Ideses and Mori can render the independent claims obvious because neither reference teaches turning features on or off during the classification stage, based on the executable being classified.

The most significant problem with Ground 3 is that, as explained in detail above, neither Ideses nor Mori teaches, suggests, or renders obvious the 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. Petitioner's Ground 3 argument does not address this deficiency in any way, let alone provide any explanation for why a skilled artisan would have found this limitation obvious in light of both references. Thus, Petitioner has failed to show that the combination of Ideses and Mori would have rendered the claimed idea obvious.

IV. THERE ARE OBJECTIVE INDICIA OF NONOBVIOUSNESS

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 23-26), 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, ¶ 98.) 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 113-118.) Petitioner, for example, markets and provides the Intercept X platform, the components of which practice the systems and methods claimed in the '844 patent.

Accordingly, secondary objective indicia of non-obviousness confirm that Petitioner cannot show a reasonable likelihood of success as to any of the challenged independent claims.

V. CONCLUSION

For the foregoing reasons, the Board should decline to institute an IPR of the '844 patent.

Date: May 16, 2023

Respectfully submitted,

KING & SPALDING LLP

BY: /s/ Brian Eutermoser

Brian Eutermoser
Registration No. 64,058

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 11,164 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.2202.

Date: May 16, 2023

Respectfully submitted,

KING & SPALDING LLP

BY: /s/ 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 2001-2016** 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:

Ryan Iwahashi
(Reg. No. 63,378)
GIBSON, DUNN & CRUTCHER LLP
1881 Page Mill Road
Palo Alto, CA 94304-1211
Tel: (650) 849-5367
RIwahashi@gibsondunn.com

*Counsel for Petitioners
Sophos Ltd. and Sophos Inc.*

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

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

Timothy P. Best (*pro hac vice*)
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 *Inter Partes* Review
Patent No. 10,599,844

Date: May 16, 2023

KING & SPALDING LLP

BY: /s/ Brian Eutermoser

Brian Eutermoser

Registration No. 64,058

Attorney for Patent Owner

Open Text Inc.