

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PALO ALTO NETWORKS, INC.
Petitioner

v.

CROGA INNOVATIONS LTD.
Patent Owner

Case IPR2024-01421
U.S. Patent No. 11,223,601

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 11,223,601**

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.	Mandatory Notices.....	3
A.	Real Parties in Interest	3
B.	Related Matters	3
C.	Lead and Backup Counsel.....	3
D.	Service Information	4
III.	Grounds for Standing.....	4
IV.	The '601 Patent.....	4
A.	Overview of the '601 Patent	4
B.	Applicant Admitted Prior Art.....	7
C.	Prosecution History.....	8
D.	Level of Ordinary Skill in the Art	11
E.	Claim Construction	11
V.	Identification of the Challenge (37 C.F.R. § 42.104(b))	12
A.	Statutory Grounds for the Challenge	12
B.	The Office has not previously considered the art applied here.	15
VI.	Grounds of Unpatentability	15
A.	<u>Ground 1</u> : Claims 1-4, 7, 9-12, and 15 are obvious over Walsh, AAPA, and Innes.	15
1.	A POSA would have been motivated to combine Walsh with Innes based on Walsh's disclosure.	15
2.	A POSA would have been motivated to combine Walsh-Innes with AAPA.....	21

***Inter Partes Review of
U.S. Patent No. 11,223,601***

3.	Independent claims 1 and 9.....	22
4.	Claims 2 and 10	36
5.	Claims 3 and 11	37
6.	Claims 4 and 12	38
7.	Claims 7 and 15	39
B.	<u>Ground 2:</u> Claims 5, 6, 13, and 14 are obvious over Walsh, AAPA, Innes, and Netfilter.	41
1.	A POSA would have been motivated to combine Walsh, AAPA, and Innes with Netfilter based on Walsh’s disclosure.....	41
2.	Claims 5 and 13	42
3.	Claims 6 and 14	43
C.	<u>Ground 3:</u> Claims 8 and 16 are obvious over Walsh, AAPA, Innes, and Franco.....	44
1.	A POSA would have been motivated to combine Walsh, AAPA, and Innes with Franco based on Walsh’s disclosure.....	44
2.	Claims 8 and 16	46
D.	<u>Ground 4:</u> Claims 1-4, 7, 9-12, and 15 are obvious over Austin, AAPA, and Bloch.	48
1.	A POSA would have been motivated to combine Austin with Bloch based on Austin’s disclosure.....	48
2.	A POSA would have been motivated to combine Austin- Bloch with AAPA.....	56
3.	Independent claims 1 and 9.....	58
4.	Claims 2 and 10	72
5.	Claims 3 and 11	73

***Inter Partes Review of
U.S. Patent No. 11,223,601***

6.	Claims 4 and 12	74
7.	Claims 7 and 15	75
E.	<u>Ground 5</u> : Claims 5, 6, 13, and 14 are obvious over Austin, AAPA, Bloch, and Walker.....	76
1.	A POSA would have been motivated to combine Austin, AAPA, and Bloch with Walker based on Austin’s disclosure.....	76
2.	Claims 5 and 13	78
3.	Claims 6 and 14	79
F.	<u>Ground 6</u> : Claims 8 and 16 are obvious over Austin, AAPA, Bloch, and Franco	81
1.	A POSA would have been motivated to combine Austin, AAPA, and Bloch with Franco based on Austin’s disclosure.....	81
2.	Claims 8 and 16	83
VII.	No Objective Indicia of Patentability	85
VIII.	The Board should not discretionarily deny the Petition.	85
A.	<i>Advanced Bionics</i> strongly favors institution.	85
B.	<i>General Plastic</i> strongly favors institution.	85
C.	<i>Fintiv</i> strongly favors institution.	86
IX.	Conclusion.....	88

PETITIONER’S EXHIBIT LIST

Exhibit	Description
1001	U.S. Patent No. 11,223,601 to Coleman et al.
1002	U.S. Patent Prosecution History of U.S. Patent No. 11,223,601
1003	Declaration of Dr. Samrat Bhattacharjee
1004	Curriculum Vitae of Dr. Samrat Bhattacharjee
1005	U.S. Publication No. 2016/0306964 to Austin et al.
1006	U.S. Patent No. 8,640,187 to Walsh
1007	U.S. Patent No. 7,849,502 to Bloch et al.
1008	U.S. Publication No. 2016/0149863 to Walker et al.
1009	U.S. Patent No. 8,161,563 to Franco et al. (“Franco”)
1010	“Productivity Tools,” Cisco WebEx (2016)
1011	“Sharing and Presenting Tools,” Cisco WebEx (2016)
1012	U.S. Patent No. 9,154,488 to Innes et al.
1013	“What is netfilter.org?,” Netfilter (2016)
1014	Declaration of Nathaniel E. Frank-White regarding the authenticity and public accessibility of WebEx, Netfilter, and X2Go
1015	Kamp, P. & Watson, R., “Jails: Confining the omnipotent root.,” <i>The FreeBSD Project</i> (2000)
1016	Bui, T., “Analysis of Docker Security,” <i>Aalto University School of Science</i> (Jan. 13, 2015)
1017	Meggers, J., “Ringin in 2015, the Webex Way,” <i>Cisco</i> (Jan. 14, 2015), accessed at https://blogs.cisco.com/collaboration/ringing-in-2015-the-webex-way#:~:text=3%20min%20read,Collaboration%20Applications%20team%20at%20Cisco.
1018	“Getting Started Guide: Cisco WebEx Meetings,” <i>Cisco WebEx</i> , (2014)
1019	“Sharing and Presenting Tools Event Center,” <i>Cisco WebEx University</i> (2014)

***Inter Partes Review of
U.S. Patent No. 11,223,601***

1020	Forouzan, B.A., “Data Communications and Networking” 5th ed. (2013)
1021	Love, R., “Linux Kernel Development,” 3d ed. (2010)
1022	John L. Hennessy & David A. Patterson, Computer Architecture (5th ed. 2012)
1023	The Associated Press, “Cisco to Buy an Online Conferencing Company,” <i>N.Y. Times</i> (Mar. 16, 2007), accessed at https://www.nytimes.com/2007/03/16/technology/16cisco.html
1024	“GoToMeeting User Guide,” Cisco (2013)
1025	Vomel, S. & Freiling, F.C., “A Survey of Main Memory Acquisition and Analysis Techniques for the Windows Operating System,” <i>Digital Investigation</i> , 1(8):3-22 (2011)
1026	McKeen, F., et al., “Innovative Instructions and Software Model for Isolated Execution,” <i>Intel Corporation</i> (2013)
1027	Wahbe, R., et al., “Efficient Software-Based Fault Isolation,” <i>SOSP '93: Proceedings of the fourteenth ACM Symposium on Operating Systems Principles</i> , pp. 203-216 (1993)
1028	Felter, W., et al., “An Updated Performance Comparison of Virtual Machines and Linux Containers,” <i>IBM</i> (Jul. 21, 2014)
1029	Price, D. & Tucker, A., “Solaris Zones: Operating System Support for Consolidating Commercial Workloads,” <i>Sun Microsystems, Inc.</i> (2004)
1030	Russinovich, M.E. & Solomon, D.E., “Microsoft Windows Internals,” 4th ed. (2005)
1031	Smalley, S., “Configuring the SELinux Policy,” <i>National Security Agency</i> (Feb. 2005)
1032	U.S. Patent No. 8,566,398 to Barnier et al. (“Barnier”)
1033	“Cisco WebEx Meetings Product Overview,” Cisco (Apr. 2015)
1034	Azab, A.M., et al., “SICE: A Hardware-Level Strongly Isolated Computing Environment for x86 Multi-Core Platforms,” <i>Association for Computing Machinery</i> (2011)
1035	<i>Croga Innovations Ltd. v. Palo Alto Networks, Inc.</i> , 2:24-cv-00208 (E.D. Tex.), Scheduling Order

I. INTRODUCTION

Palo Alto Networks, Inc. petitions for *inter partes* review of claims 1-16 (“challenged claims”) of U.S. Patent No. 11,223,601 (EX1001, “’601 patent”). The Board should institute this proceeding because the ’601 patent claims nothing more than well-known techniques developed long before the ’601 patent’s earliest possible priority date.

The challenged claims are directed to two simple techniques. They first recite the use of an isolated computing environment to protect “*multi-user collaboration software*” (e.g., WebEx) from malware or other untrusted content. The claims then recite the use of a proxy server to authenticate the isolated computing environment when accessing content over a network.

Both of these claimed techniques were known long before the earliest possible priority date of the ’601 patent. Like the first claimed technique, Walsh (EX1006) and Austin (EX1005) each disclose isolated computing environments that use the host operating system. And the ’601 patent admits that the claimed “*multi-user collaboration software*” is met by WebEx, which was commercially available in the prior art. Like the second claimed technique, Innes (EX1012) and Bloch (EX1007) each disclose the use of a proxy appliance to authenticate a computer resource over a network. The other relied-on references show that the dependent claims merely recite trivial—and obvious—additional limitations.

***Inter Partes Review of
U.S. Patent No. 11,223,601***

A person of ordinary skill in the art (“POSA”) would have been motivated to combine Walsh with Innes, and Austin with Bloch, with a reasonable expectation of success. Walsh’s content isolation is described in a Linux operating-system (OS) environment, and Innes discloses techniques for authenticating computer resources over a network in such an environment. Similarly, Austin’s content isolation is described in a Microsoft OS environment, and Bloch discloses techniques for authenticating computer resources over a network in such an environment. Using Innes’s and Bloch’s techniques in the systems of Walsh and Austin, respectively, would have amounted to nothing more than combining known prior-art elements (Walsh’s and Austin’s content isolation) with known methods (Innes’s and Bloch’s authentication techniques) to yield predictable results. The claims thus would have been obvious over the following combinations.

Ground	Basis	References	Claims
1	§103	Walsh and Innes	1-4, 7, 9-12, 15
2	§103	Walsh, Innes, and Netfilter	5, 6, 13, 14
3	§103	Walsh, Innes, and Franco	8, 16
4	§103	Austin and Bloch	1-4, 7, 9-12, 15
5	§103	Austin, Bloch, and Walker	5, 6, 13, 14
6	§103	Austin, Bloch, and Franco	8, 16

None of the prior art relied on here was previously considered by, or presented to, the Office with respect to the ’601 patent. The Board should institute this proceeding and find the challenged claims unpatentable.

II. MANDATORY NOTICES

A. Real Parties in Interest

The real party-in-interest is Palo Alto Networks, Inc. (“PAN”).

B. Related Matters

The ’601 patent is asserted against PAN in *Croga Innovations Ltd. v. Palo Alto Networks, Inc.*, 2:24-cv-00208 (E.D. Tex.), filed Mar. 22, 2024.

The ’601 patent is also asserted against Cisco Systems, Inc. (“Cisco”) in *Croga Innovations Ltd. v. Cisco Systems, Inc.*, 2:24-cv-00065 (E.D. Tex.), filed Feb. 1, 2024.

On August 13, 2024, Cisco filed an *inter partes* review (“IPR”) petition of the ’601 patent. PAN was not involved in Cisco’s IPR petition. And that IPR petition did not identify or assert any of the prior art that is asserted here.

C. Lead and Backup Counsel

Lead Counsel

Jonathan Tuminaro, Reg. No. 61,327
STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.
1101 K St. N.W., 10th Floor
Washington, DC 20005
Email: jtuminar-PTAB@sternekessler.com
Tel.: 202.772.8967
Fax: 202.371.2540

Backup Counsel

Ethan Goldschen, Reg. No. 80,057
STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.
1101 K St. N.W., 10th Floor
Washington, DC 20005
Email: egoldschen-PTAB@sternekessler.com
Tel.: 202.772.1296
Fax: 202.371.2540

Daniel S. Block, Reg. No. 68,395
STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.
1101 K St. N.W., 10th Floor
Washington, DC 20005
Email: dblock-PTAB@sternekessler.com
Tel.: 202.772.8735
Fax: 202.371.2540

D. Service Information

Please address all correspondence to lead and backup counsel at the email addresses shown above and PTAB@sternekessler.com. Petitioner consents to electronic service by email.

III. GROUNDS FOR STANDING

PAN certifies the '601 patent is available for IPR and that it is not barred or estopped from requesting IPR of the '601 patent challenging the patent claims on the grounds identified in this Petition.

IV. THE '601 PATENT

A. Overview of the '601 Patent

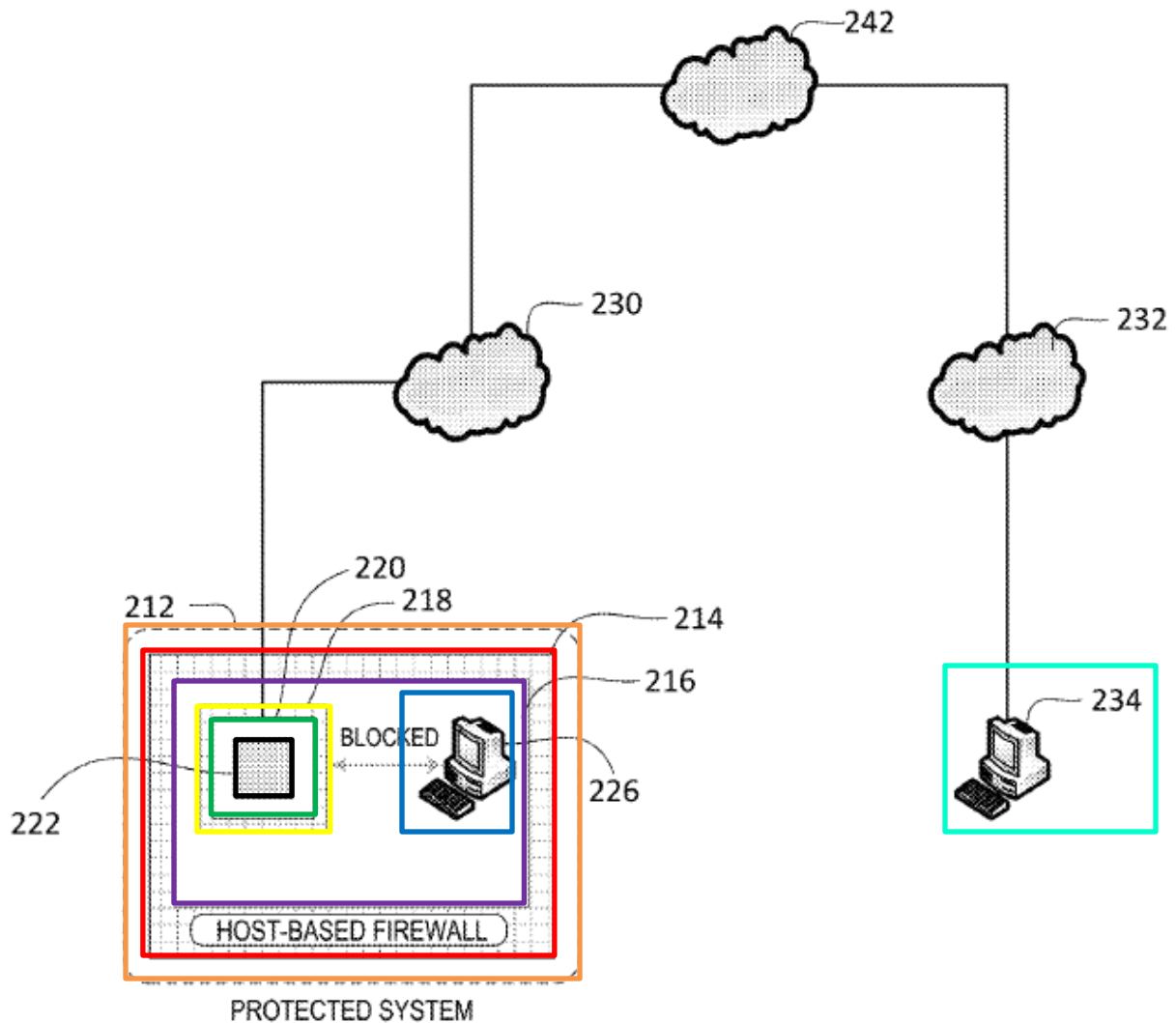
The '601 patent describes a system for “isolation of collaboration software on a host computer system.” EX1001, Abstract. This system uses “a workspace” and an “isolated computing environment.” EX1001, 1:62-65.

Trusted applications run in “the workspace via a first memory space (e.g., a trusted memory space).” EX1001, 1:62-64. Untrusted applications, or applications that are to be “isolated,” run in “the isolated computing environment via a second

memory space.” EX1001, 2:8-9. The isolated computing environment isolates the workspace by “an internal isolation firewall,” which prevents “communication between the isolated computing environment and the workspace.” EX1001, 1:65-2:2. The host computer system is “configured to run a collaboration software . . . that enables interaction with one or more other host computer systems.” EX1001, 3:4-6. The collaboration software runs “within the untrusted memory space.” EX1001, 3:7-8.

The workspace includes “one or more processes operating within the operating system that are not restricted by the sandbox container process.” EX1001, 5:44-46. The operating system includes resources enabling operation of both the workspace and the sandbox container. EX1001, 5:46-54. The applications within the sandboxed computing environment are permitted “to access untrusted network destinations via an intermediate connection device [e.g., a web proxy and/or an authentication device].” EX1001, 6:9-14.

Figure 2 (annotated below) presents an embodiment of the alleged invention. EX1001, FIG. 2. “The host computer system 212 [orange below] may include a host-based firewall 214 [red below], an operating system 226 [blue below], a sandboxed computing environment (e.g., an isolated computing environment) 220 [green below], an internal isolation firewall 218 [yellow below], and a workspace 216 [purple below].” EX1001, 9:62-10:3.



EX1001, FIG. 2 (annotated).

Collaboration software 222 (black above) enables the host computer to participate in “software and/or web-based meeting[s].” EX1001, 11:43-45. The collaboration software allows another device to “access, view, interact with, download from, upload to, and/or modify” an application on the host. EX1001, 2:19-23.

The collaboration software operates within “sandboxed computing environment 220,” restricting the other devices’ abilities to access the trusted memory (e.g., the workspace) of the host computer. EX1001, 11:65-12:3. For example, “[i]f malware is transferred to the host computer system 212 due to interactions with the host computer system 234 [(teal above)], the malware may be isolated to the sandboxed computing environment 220.” EX1001, 12:25-30.

B. Applicant Admitted Prior Art

The ’601 patent admits that the claimed multi-user interactive software application “may be implemented using software that is commercially available, for example, such as WebEx, GoToMeeting, etc.” EX1001, 4:45-47. This applicant admitted prior art (“AAPA”) shows that multi-user, interactive software applications were known and, as explained below, would have been obvious to incorporate into systems that perform content isolation. *See Qualcomm Inc. v. Apple Inc.*, 24 F.4th 1367, 1375-76 (Fed. Cir. 2022) (holding that AAPA can be used in an IPR to show obviousness, provide motivation to combine, or supply missing claim limitations); June 9, 2022 Memorandum from Katherine K. Vidal, “Updated Guidance on the Treatment of Statements of the Applicant in the Challenged Patent in *Inter Partes* Reviews under § 311,” at 4 (noting that “an IPR petition” that “relies on admissions in combination with reliance on one or more

prior art patents or printed publications . . . must be considered by the Board in its patentability analysis”).

C. Prosecution History

The '601 patent was rejected three times during prosecution.

First Office Action. The first Office Action included several rejections, including, as relevant here, a rejection of application claims 1-16 under 35 U.S.C. § 103 over U.S. Patent No. 8,966,464 (“Christopher”) in view of U.S. Publication No. 2014/0040979 (“Barton”). EX1002, 0103-07. The Examiner cited Christopher as disclosing: (1) a workspace; (2) an isolated computing environment; and (3) an internal isolation firewall. EX1002, 0103-04. The Examiner cited Barton as disclosing multi-user, interactive software communicating with an untrusted destination via a proxy device. EX1002, 0104.

In response to the obviousness rejection, the Applicant filed an amendment and remarks. Specifically, Applicant amended independent claims 1 and 9 to recite the system sending data “to an untrusted destination from the multi-user interactive software application associated with the isolated computing environment via a proxy device.” EX1002, 0132. And the Applicant argued the “managed . . . applications” in Barton cannot send data to untrusted destinations because the managed applications are prohibited from communicating with “unmanaged application[s].” EX1002, 0139.

Second Office Action. The Examiner issued a Final Rejection with the same § 103 rejection as in the First Office Action, concluding that Barton discloses managed applications may communicate with unmanaged applications (i.e., untrusted destinations). EX1002, 0147-51.

The Applicant filed an amendment and a Request for Continued Examination. The Applicant narrowed independent claims 1 and 9 by adding the following element: “authentica[ting] the isolated computing environment with an authentication device.” EX1002, 0166. The Applicant further amended the sending step to “send data to an untrusted destination from the multi-user interactive software application via a proxy device when the isolated computing environment has been authenticated.” EX1002, 0166-68. The Applicant again argued that: (1) Christopher fails to send data to an untrusted application; and (2) in Barton, managed apps (e.g., trusted apps) cannot send data to unmanaged apps (e.g., untrusted apps). EX1002, 0171.

Third Office Action. The Examiner issued a second non-final office action, which, as relevant here, rejected claims 1-16 under § 103 over “SICE: A Hardware-Level Strongly Isolated Computing Environment for x86 Multi-core Platforms” (“Azab”) in view of U.S. Patent No. 8,566,398 (“Barnier”). EX1002, 0183. Azab discloses “a novel framework to provide hardware-level isolation . . . [that] does not rely on any software component in the host environment (i.e., an OS

or a hypervisor). EX1034, Abstract. The isolation is implemented via, “hardware, the BIOS, and the System Management Mode (SMM).” *Id.* Barnier discloses an extranet including “non-related entities” that may share applications and data through the extranet connection. EX1032, 3:41-45.

In response, the Applicant narrowed both the claimed “*workspace*” and the “*isolated computing environment*” to use a “*host operating system.*” EX1002, 0199, 0201.

implement a workspace that uses a host operating system, wherein the workspace enables ~~is configured to enable~~ operation of a first set of one or more applications or processes via a first memory space;

implement an isolated computing environment that uses the host operating system, wherein the isolated computing environment enables ~~being configured to enable~~ operation of a second set of one or more applications or processes via a second memory space, and wherein the second set of one or more applications or processes comprises a multi-user interactive software application;

EX1002, 0199.

The Applicant argued that Azab’s isolated environment doesn’t use the host operating system because Azab states that it doesn’t rely on software in the host environment. EX1002, 0205. The applicant further argued that Barnier includes no discussion of operating systems. *Id.*

Claims 1-16 were allowed and claim 17 was canceled in an *Ex Parte* Quayle action. The notice stated that the prior art supposedly did not teach the claim

limitation of “*an isolated computing environment that uses the host operating system.*” EX1002, 0240. But, as explained below, this limitation is taught by both Walsh and Austin.

D. Level of Ordinary Skill in the Art

A POSA in the technical field of the '601 patent at the relevant time would have had a B.S. degree in Computer Science, Computer Engineering, or an equivalent field, as well as at least 2-3 years of academic or industry experience in the design, analysis, and implementation of operating systems, as well as the design, analysis, and monitoring of computer networks, including issues of network security and network administration, or comparable industry experience. EX1003, ¶30.

Petitioner's expert, Dr. Samrat Bhattacharjee, has more than ordinary skill in the art. EX1003, ¶¶5-17. His opinions are supported by not only his decades of experience, but also the references cited in this petition and the additional reasoning and documents cited in his declaration. EX1003, ¶¶4-16.

E. Claim Construction

No claim term needs an explicit construction. *See Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999). PAN has interpreted the terms according to their ordinary and customary meanings as understood by a

POSA in view of the specification. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2005).

V. IDENTIFICATION OF THE CHALLENGE (37 C.F.R. § 42.104(B))

A. Statutory Grounds for the Challenge

This Petition relies on the following statutory grounds and combination of references.

Ground	Basis	References	Claims
1	§103	Walsh and Innes	1-4, 7, 9-12, 15
2	§103	Walsh, Innes, and Netfilter	5, 6, 13, 14
3	§103	Walsh, Innes, and Franco	8, 16
4	§103	Austin and Bloch	1-4, 7, 9-12, 15
5	§103	Austin, Bloch, and Walker	5, 6, 13, 14
6	§103	Austin, Bloch, and Franco	8, 16

These references all qualify as prior art. The '601 patent was filed September 25, 2018. EX1001, (22). It claims priority to a provisional application filed September 28, 2017. EX1001, (60). Petitioner does not concede that the challenged claims are entitled to the provisional application's filing date. Nonetheless, all references relied on here qualify as prior art even if it is.

Walsh: U.S. Patent No. 8,640,187 to Walsh describes methods for creating an isolated execution environment to access untrusted content “without exposing other processes and resources of the computing system to the untrusted content.”

EX1006, Abstract. Walsh issued on January 28, 2014 and qualifies as prior art under § 102(a)(1). EX1006, (45).

Innes: U.S. Patent No. 9,154,488 to Innes et al. describes methods for authenticating and accessing remote resources using a proxy device. EX1012, Abstract. Innes issued on October 6, 2015 and qualifies as prior art under § 102(a)(1). EX1012, (45).

Netfilter: “The netfilter.org project” (EX1013, “Netfilter”) was published on the Internet in 2016 by Netfilter. Its public availability is confirmed by an Internet Archive WayBack Machine capture of the publication in its entirety on the netfilter.org website at least as early as January, 2016. EX1014, 0010-12. EX1013 qualifies as prior art under §102(a)(1).

Franco: U.S. Patent No. 8,161,563 to Franco et al. (EX1009, “Franco”) describes methods for executing an Internet-related application “with a reduced privilege level” to “prohibit the application from accessing portions of an associated computing device.” EX1009, Abstract. Franco issued on April 17, 2012 and qualifies as prior art under §102(a)(1). EX1009, (45).

Austin: U.S. Publication No. 2016/0306964 to Austin et al. describes methods for isolating untrusted content in a secondary user account that does not have access to a primary user account. EX1005, Abstract. Austin was published on October 20, 2016 and qualifies as prior art under § 102(a)(1). EX1005, (43).

Bloch: U.S. Patent No. 7,849,502 to Bloch et al. describes methods for using a proxy device and authentication server to provide integrated authentication services. EX1007, 8:16-24, FIG. 6B. Bloch issued on December 7, 2010 and qualifies as prior art under § 102(a)(1). EX1007, (45).

Walker: U.S. Publication No. 2016/0149863 to Walker et al. (EX1008, “Walker”) describes methods for managing a firewall on a host computing device. EX1008, Abstract. Walker published on May 26, 2016 and qualifies as prior art under § 102(a)(1). EX1008, (43).

WebEx: This petition relies on two papers describing the functionality of Cisco’s WebEx product. These papers are collectively referred to as “WebEx”:

- Cisco WebEx, *Productivity Tools* (2016) (EX1010) was published on the Internet by Cisco. Its public availability is confirmed by an Internet Archive WayBack Machine capture of the publication in its entirety on the Cisco website at least as early as April 23, 2016. EX1014, 0015-17. Exhibit 1010 qualifies as prior art under § 102(a)(1).
- Cisco WebEx, *Sharing and Presentation Tools* (2016) (EX1011) was published on the Internet by Cisco. Its public availability is confirmed by an Internet Archive WayBack Machine capture of the publication in its entirety on the Cisco website at least as early as April 23, 2016.

EX1014, 0019-22. Exhibit 1011 qualifies as prior art under §
102(a)(1).

This Petition is supported by the testimony of PAN's technical expert, Dr. Samrat Bhattacharjee, who is a tenured professor in the Department of Computer Science and the Institute for Advanced Computer Studies at the University of Maryland with over 25 years of academic and research experience in the design and implementation of network-security technology.

B. The Office has not previously considered the art applied here.

None of the applied references were previously considered by the Office. During prosecution, neither the patentee nor the examiner cited or applied any of the prior art relied on in this petition. Nor were any of the applied references cited in Cisco's IPR petition against the '601 patent. Thus, the same or substantially the same prior art or arguments were not previously presented to the Office.

VI. GROUNDS OF UNPATENTABILITY

A. Ground 1: Claims 1-4, 7, 9-12, and 15 are obvious over Walsh, AAPA, and Innes.

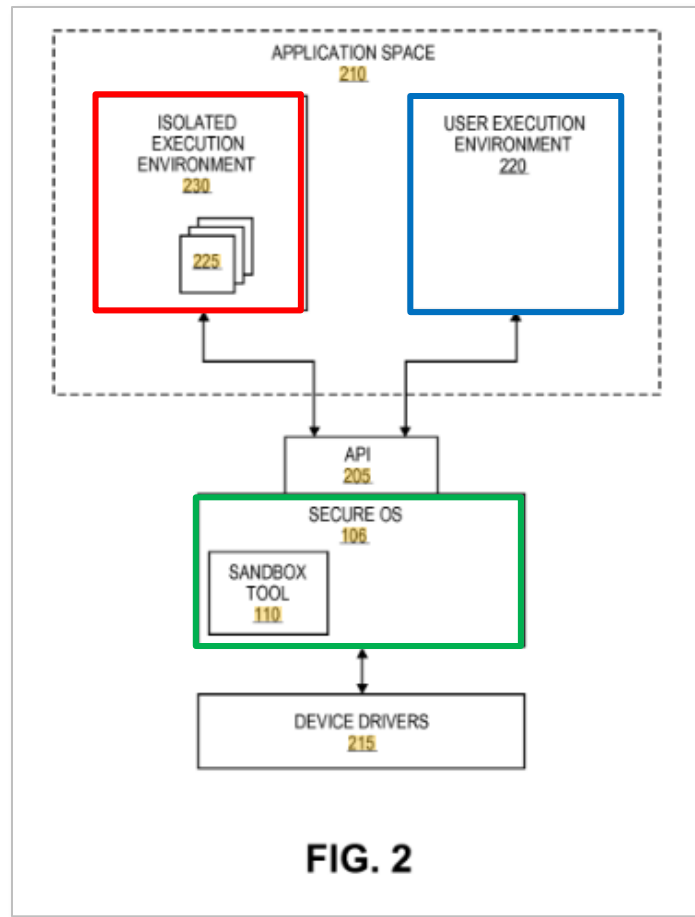
1. A POSA would have been motivated to combine Walsh with Innes based on Walsh's disclosure.

Walsh is directed to "computer security" in "distributed computing environments." EX1006, 1:8-14. In these environments, Walsh acknowledges that "malicious entities can plant viruses, Trojans, or other malicious agents in publicly

available content in order to attack the users' computing systems and steal sensitive information from the users." EX1006, 1:19-23.

To address these issues, Walsh discloses techniques for accessing untrusted content such as "programs" and "applications" "without exposing a computing system to untrusted content and possibly malicious content." EX1006, 2:32-36. Specifically, Walsh provides "an isolated execution environment," which uses "a secure operating system (OS), such as security enhanced LINUX (SELinux)," to access the untrusted content. EX1006, 2:36-44.

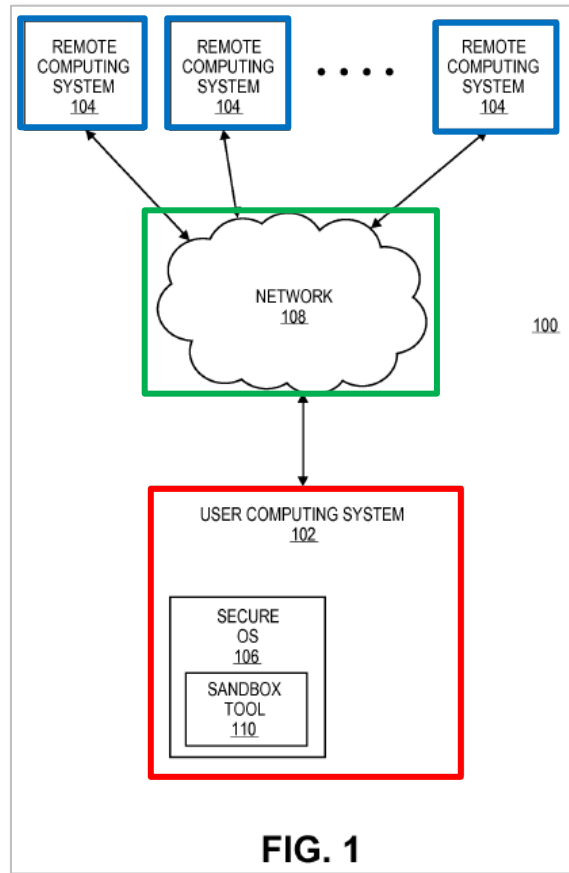
Exemplary software for providing such content isolation is illustrated in Walsh's Figure 2 (annotated below). "The content," Walsh explains, can be accessed over the Internet via a "web browser" application. EX1006, 3:67-4:4. Trusted content is executed in a "user execution environment 220" (in blue). EX1006, 5:30-32. Untrusted content, in contrast, is executed in an "isolated execution environment 230" (in red) "without exposing other processes such as the user execution environment 220" to the untrusted content. EX1006, 5:50-57. The "user execution environment 220" and the "isolated execution environment 230" both use a "secure OS" (in green), but each are provided with their own memory "namespace" so that "the processes of isolated execution environment 230 cannot access files and processes of any other namespace." EX1006, 6:23-33.



EX1006, FIG. 2 (annotated).

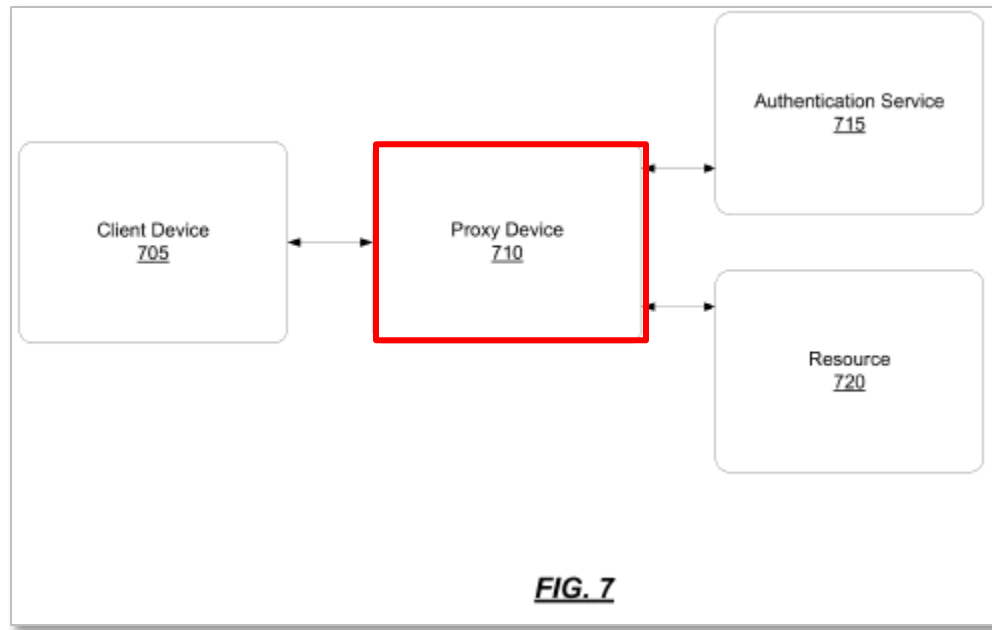
A POSA would have been motivated to combine Walsh with Innes based on the teachings in Walsh itself. EX1003, ¶¶136-148. Referring to annotated Figure 1 (below), Walsh discloses that a “user computing system 102” (in red) is connected to other “remote computing systems 104” (in blue) via “network 108” (in green). EX1006, 3:24-54. Walsh says that the “user computing system 102” can access content via “network 108” using “a client-server model.” EX1006, 4:66-5:3. To the extent that Walsh does not provide details about the client-server model, a POSA

would have been motivated to combine Walsh with a reference, like Innes, that has those details. EX1003, ¶141.



EX1006, FIG. 1 (annotated).

Like Walsh, Innes discloses that remote computers can access content via a network “using a web browser.” EX1012, 4:27-47, FIG. 1. The remote computers “may wish to access enterprise resources that require authentication.” EX1012, 23:43-45. To provide such authentication, Walsh discloses a “proxy device 710” (in red below) that can provide “authentication service 715” to a “client device 705”:



EX1012, FIG. 7 (annotated).

And Innes discloses “example method steps for authentication” and for “providing secured access to resources using a proxy,” such as “proxy device 710.” EX1012, 27:43-45, FIGs. 9A-C.

A POSA would have been further motivated to implement Walsh’s user computing system on a network including Innes’s proxy device and authentication services to improve computer and network security. EX1003, ¶145. Walsh recognizes that content may be insecure and untrustworthy. EX1003, ¶146; EX1006, 4:21-22. To provide protection, Walsh’s computing devices implement content isolation by using a “user execution environment 220” (for trusted content) and an “isolated execution environment 230” (for untrusted content). EX1006, FIG. 2.

To provide additional protection, a POSA would have been motivated to place the computing device on a network with Innes's proxy device. EX1003, ¶147. As noted above, Walsh's computing devices access the trusted/untrusted content over a network using a web browser. EX1006, 3:64-4:4. The content, Innes teaches, can be "securely" accessed "using a proxy device." EX1012, 1:6-8. For example, Innes's proxy device is configured to establish an authenticated session with the requested resource and provide session information to the computing device. EX1012, 22:15-18. The computing device uses the session information to verify the authentication session between the proxy device and resource. EX1012, 22:37-40. Combining Innes's authentication techniques with Walsh's computing device would have provided an additional layer of security for accessing the trusted/untrusted content over the network. EX1003, ¶147.

In addition, the combination of Walsh and Innes would have been obvious to try. EX1003, ¶177. In the client-server context disclosed by Walsh, EX1006, 4:66-5:3, machines can access the Internet using known paradigms, for example, using a network-address-translation device or using a proxy. EX1003, ¶177. Thus, it would have been obvious for a POSA to try to connect Walsh's computing system to the Internet via a proxy device as disclosed by Innes. *Id.*

A POSA would have had a reasonable expectation of success combining Walsh with Innes because the combination would amount to nothing more than

placing Walsh's computing device on a network with Innes's proxy device.

EX1003, ¶148. Walsh's computing device is a standard computing system.

EX1006, 3:42-46. It communicates and shares data via a network that can be wired or wireless. EX1006, 3:57-63. Innes's proxy device communicates with a client device (e.g., Walsh's computing device) over a network to provide resource access.

EX1012, 22:13-46.

2. A POSA would have been motivated to combine Walsh-Innes with AAPA.

The '601 patent admits that the claimed "*multi-user interactive software application*" can be implemented via commercially available products such as WebEx. EX1001, 4:43-47. WebEx was a well-known product and was used as a standalone application, an email client plugin, or within a web browser. EX1003, ¶¶149-150.

A POSA would have been motivated, with a reasonable expectation of success, to use Innes's proxy appliance with Walsh's client computer in order to "securely" access WebEx traffic via Walsh's computer system. EX1003, ¶150; EX1012, 1:6-8. Walsh explains that a user computer system may access content over the Internet via a "web browser" application. EX1006, 3:67-4:4. A POSA would have understood that WebEx was a well-known application for accessing traffic over the Internet, and a POSA would have understood that WebEx would be (i) installed as a standalone program on Walsh's computer system or (ii) used

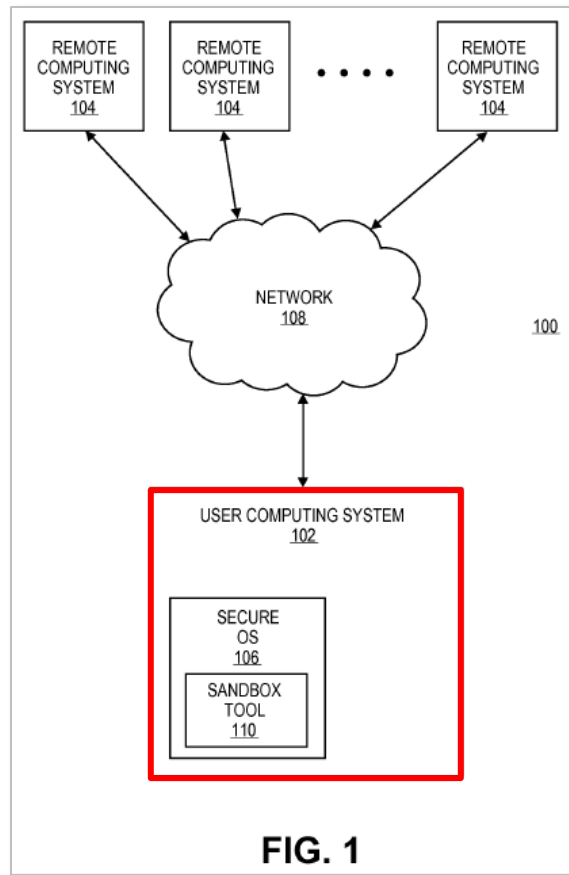
within a web browser or email client on Walsh's computer system. EX1003, ¶150.

The combined Walsh-AAPA system would then be used on a network including Innes's proxy device to provide Walsh's client device with secure access to network resources (e.g., an HTTP server). EX1012, 28:34-48. A POSA would not have experienced any unreasonable technical hurdles using Walsh's computing device to access WebEx on a network including Innes's proxy device. EX1003, ¶150.

3. Independent claims 1 and 9

(a) [1-pre] "A system" and [9-pre] "A method"

To the extent the preambles are limiting, Walsh discloses them. Referring to Figure 1 (annotated below), Walsh discloses "a user computing system 102" (a "*system*") included within a network 100. EX1006, 3:35-38. Walsh's "user computing system 102" implements a "*method*" because it creates an isolated execution environment and accesses "content (trusted or untrusted) without exposing other processes and resources of the user computing system 102 to the content." EX1003, ¶151; EX1006, 4:30-35.



EX1006, FIG. 1 (annotated).

(b) [1A] “a memory”

Walsh discloses this element.

Walsh’s “user computing system 102” includes “hardware resources” such as “processors” and “**memory.**” EX1003, ¶¶152-153; EX1006, 3:38-43¹.

(c) [1B] “a processor configured to”

Walsh discloses this element.

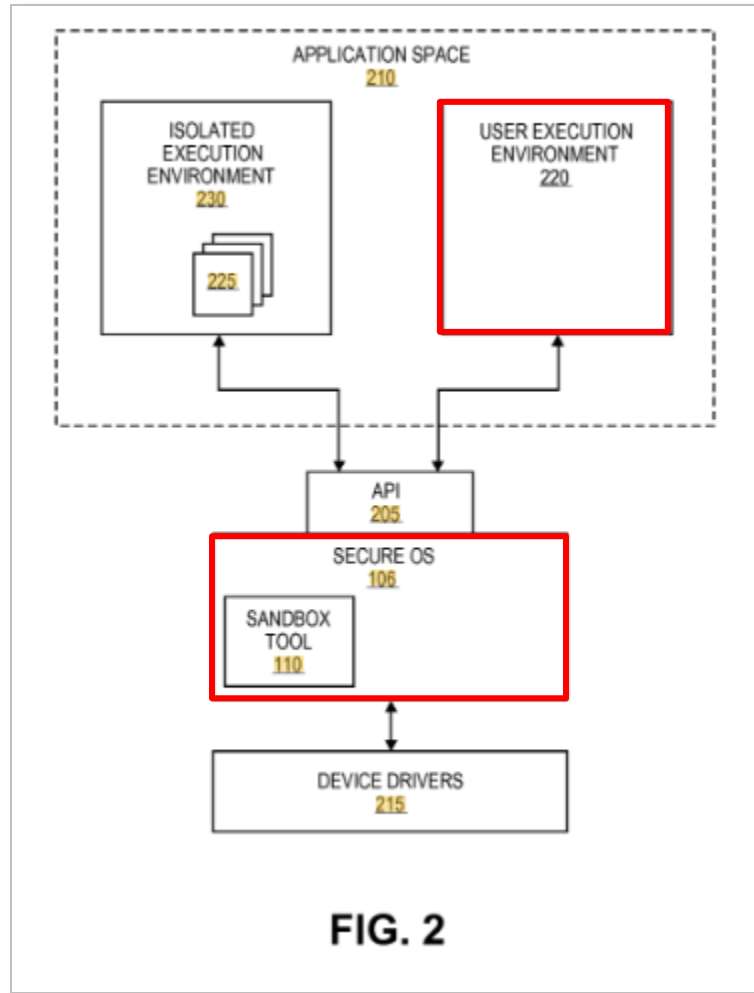
¹ Emphasis added throughout unless otherwise noted.

Walsh's "user computing system 102" includes "hardware resources," such as "**processors**" and "memory." EX1003, ¶¶154-155; EX1006, 3:38-40.

- (d) [1C] and [9A] "[implement/implementing] a workspace that uses a host operating system, wherein the workspace enables operation of a first set of one or more applications or processes via a first memory space"**

Walsh discloses this element.

Walsh's "user execution environment 220" ("*a workspace*") is created by a "secure OS 106" ("*a host operating system*") of the "user computing system 102": "secure OS 106 can be configured to create a user execution environment 220 in the application space 210." EX1006, 5:28-32; *see also* EX1006, FIG. 1. "The user execution environment 220" ("*workspace*") "*enables operation of a first set of one or more applications or processes*" because Walsh discloses that the "user execution environment 220 allows users to interact with the user computing system 102 to access content such as run application and programs, view files, etc." EX1006, 5:30-32. The "user execution environment 220" ("*workspace*") is supported by its own "user namespace such as home directory and temporary directory" ("*a first memory space*"). EX1006, 5:33-40. The "sandbox tool 110" is configured to "allocate resources (storage space, memory, etc.) of the user computing system 102, which are necessary to create the isolated execution environment." *Id.*, 4:35-38; EX1003, ¶¶156-157.



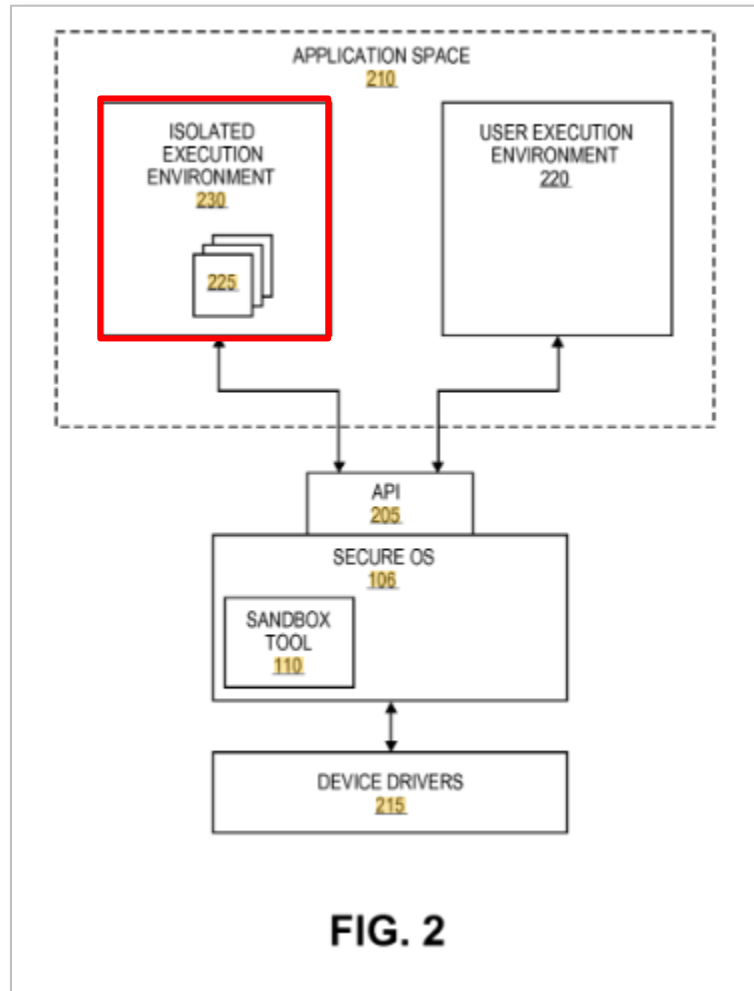
EX1006, FIG. 2 (annotated).

- (e) [1D] and [9B] “[implement/implementing] an isolated computing environment that uses the host operating system”

Walsh discloses this element.

Walsh discloses an “isolated execution environment 230” (“*isolated computing environment*”) that is created by the “secure OS 106” (“*host operating system*”): “[S]andbox tool 110 can be configured to cooperate with components of

a secure OS 106, to create an **isolated execution environment 230 . . .**” EX1006, 5:50-57; EX1003, ¶¶158-159.



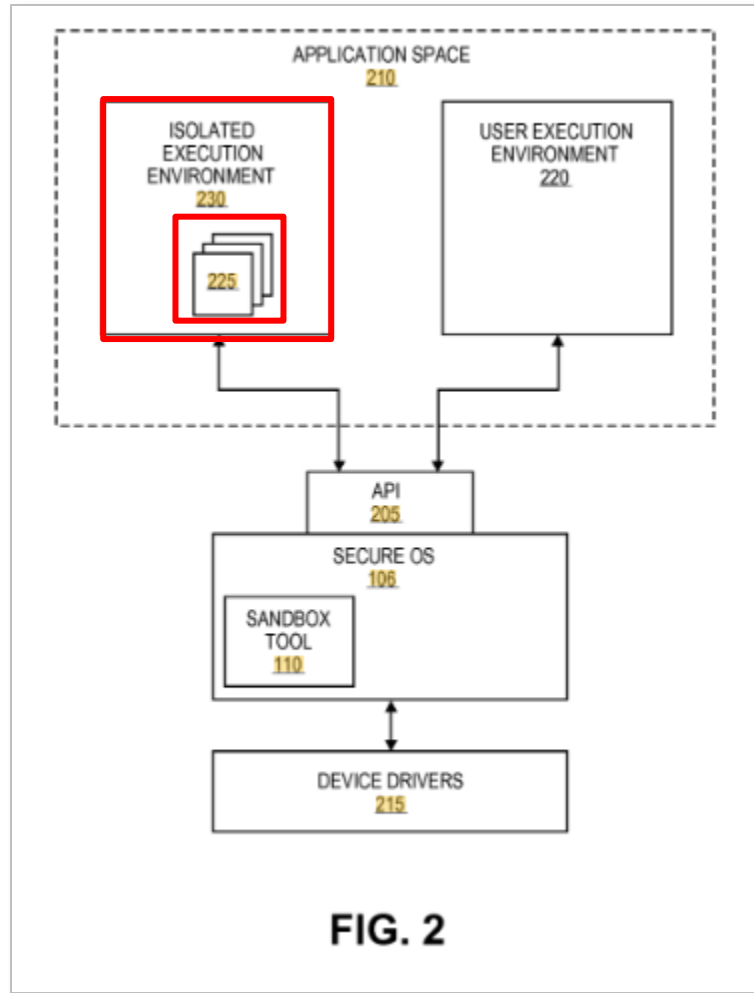
EX1006, FIG. 2 (annotated).

- (f) [1E] and [9C] “wherein the isolated computing environment enables operation of a second set of one or more applications or processes via a second memory space”

Walsh discloses this element.

Walsh's "isolated execution environment 230" (*"isolated computing environment"*) executes applications and programs (*"a second set of one or more applications or processes"*) that require isolation. EX1006, 4:47-52 (*"[S]andbox tool 110 can be configured to create the isolated execution environment to allow the **applications, programs** . . . to be . . . **executed** . . ."*), 10:18-21 (*"[C]ontent can be any content (files, **applications**, programs) which is required by the isolated execution environment 230 . . ."*), 10:21-23 (*"[C]ontent 225 can be **application programs**, files, data, etc. . . ."*).

Walsh discloses "storage space" and "memory" (*"a second memory space"*) that is allocated to the "isolated execution environment 230" by a "sandbox tool 110." EX1006, 4:35-42, 5:57-61. The "isolated execution environment 230 can only access the resources" (e.g., the "memory") allocated to it. EX1006, 5:61-65. In particular, the "sandbox tool 110" creates "a namespace for the isolated execution environment 230 (e.g., home directory and tmp directory) which is separate from the namespace of the user execution environment 220." EX1006, 6:4-9; EX1003, ¶¶160-163.



EX1006, FIG. 2 (annotated).

- (g) [1F] and [9D] “wherein the second set of one or more applications or processes comprises a multi-user interactive software application”

Walsh alone or in combination with AAPA discloses this element.

Walsh discloses that the content executing in the isolated execution environment 230 (“*the second set of one or more applications or processes*”) may include “any number and type of data, applications programs such as word processing applications, **web browser applications**, file sharing applications,

electronic mail (e-mail) applications, multimedia applications, **chat applications**, etc.” EX1006, 3:67-4:4. A chat application, as disclosed by Walsh, is a “*multi-user interactive software application*.” EX1003, ¶165.

In addition, the ’601 patent admits that the claimed “*multi-user interactive software application*” existed in commercially available products such as WebEx. EX1001, 4:45-49. And a POSA would have understood that WebEx could be used (i) within a web browser, (ii) as a standalone application, or (iii) as an email client plugin. EX1003, ¶166.

As explained above, a POSA would have been motivated, with a reasonable expectation of success, to execute WebEx (“*multi-user interactive software application*”) as one of Walsh’s untrusted applications in the isolated execution environment 230 so that WebEx could not “access files and processes of any other namespace.” EX1006, 6:23-33. This combination, moreover, would have amounted to nothing more than combining familiar elements (e.g., Walsh’s content isolation and the commercially available WebEx application) according to their known methods of use. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416 (2007).

In the Walsh-AAPA combined system, Walsh’s untrusted application (“*the second set of one or more applications or processes*”) would be WebEx (“*a multi-user interactive software application*”). EX1003, ¶¶167-168.

(h) [1G] and [9E] “[isolate/isolating] the isolated computing environment from the workspace using an internal isolation firewall”

Walsh discloses this element.

Walsh discloses a “sandbox tool 110” that implements “security policies” (*“internal isolation firewall”*) to isolate the “isolated execution environment 230” (*“isolated computing environment”*) from the “user execution environment 220” (*“workspace”*):

[S]andbox tool 110 can be configured to cooperate with components of a secure OS 106, to create an isolated execution environment 230 for accessing content 225 (trusted or untrusted) **without exposing other processes such as the user execution environment 220 and resources of the user computing system 102 to the content 225.**

EX1006, 5:50-57.

In particular, the sandbox tool 110 is “configured to apply security policies of the secure OS 106 to the isolated execution environment 230 such that **the content 225 running in the isolated execution environment 230 can only access the resources allocated to the isolated execution environment 230.**” EX1006, 5:61-65.

(i) [1H] and [9F] “[authenticate/authenticating] the isolated computing environment with an authentication device”

The Walsh-AAPA-Innes combination discloses this element.

A POSA would have been motivated to combine Walsh-AAPA with Innes to securely access content on the network disclosed by Walsh. Specifically, Walsh notes that in distributed computing environments local computing systems may be connected to remote computing systems via a network. EX1006, 3:24-54. Walsh says that the local computing systems can access content via the network using “a client-server model.” EX1006, 4:66-5:3. To the extent that Walsh does not provide implementation details about the client-server model, Innes does.

Like Walsh, Innes discloses that remote computers can access content via a network “using a web browser.” EX1012, 4:27-47, FIG. 1. The remote computers “may wish to access enterprise resources that require authentication.” EX1012, 23:43-45. To provide such authentication, Innes discloses a “proxy device 710.” And Innes discloses “example method steps for authentication” and for “providing secured access to resources using a proxy,” such as “proxy device 710.” EX1012, 27:43-44, FIGs. 9A-C.

A POSA would have been motivated, with a reasonable expectation of success, to implement the Walsh-AAPA user computing system on a network including Innes’s proxy device and authentication services in order to securely access WebEx content on Walsh’s network. EX1003, ¶176. Walsh discloses that computing devices may implement content isolation by using a “user execution environment 220” (for trusted content) and an “isolated execution environment

230” (for untrusted content). EX1006, FIG. 2. The computing devices access the trusted/untrusted content over a network using a web browser. EX1006, 3:64-4:4. WebEx is used within a web browser. EX1003, ¶176. The content, Innes teaches, can be “securely” accessed “using a proxy device.” EX1012, 1:6-8. And Innes’s “proxy device” allows computing devices to request certain authentication services, which “may be provided by an authentication server.” EX1012, 27:43-61. Moreover, this combination would have been obvious because it would amount to nothing more than combining familiar elements according to their known methods of use. *KSR*, 550 U.S. at 416.

In addition, the combination of Walsh and Innes would have been obvious to try. EX1003, ¶177. In the client-server context disclosed by Walsh, EX1006, 4:66-5:3, machines can access the Internet using known paradigms, for example, using a network-address-translation device or using a proxy. EX1003, ¶177. Thus, it would have been obvious for a POSA to try to connect Walsh’s computing system to the Internet via a proxy device as disclosed by Innes. *Id.*

In the Walsh-AAPA-Innes system, Walsh’s “isolated execution environment 230” (“*isolated computing environment*”) would be authenticated via Innes’s authentication server (“*authentication device*”).

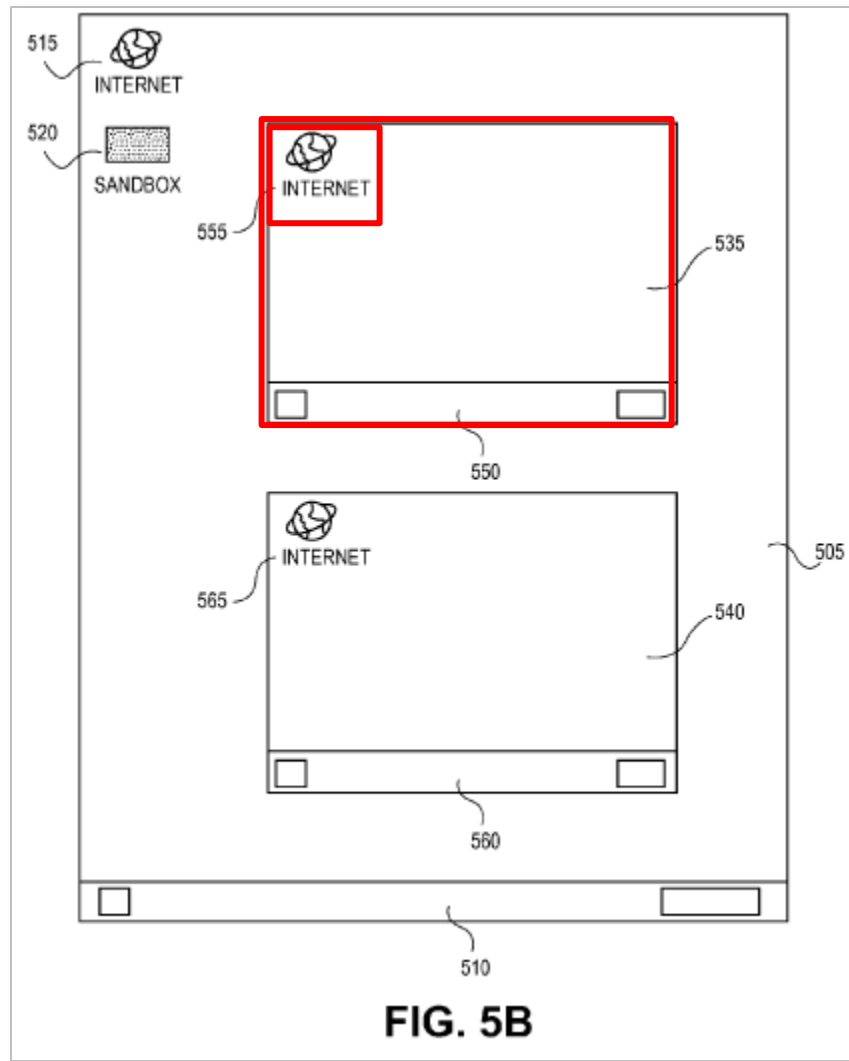
- (j) [1I] and [9G] “[send/sending] data to an untrusted destination from the multi-user interactive software application via a proxy device when the isolated computing environment has been authenticated.”

The Walsh-AAPA-Innes combination discloses this element. EX1003, ¶¶179-183.

Walsh recognizes that untrusted tasks may be accessed over the Internet using a web browsing. EX1006, 1:12-25. To address this issue, Walsh discloses that a web browser may be isolated when instructed to view (“*send data to*”) an untrusted website (“*an untrusted destination*”):

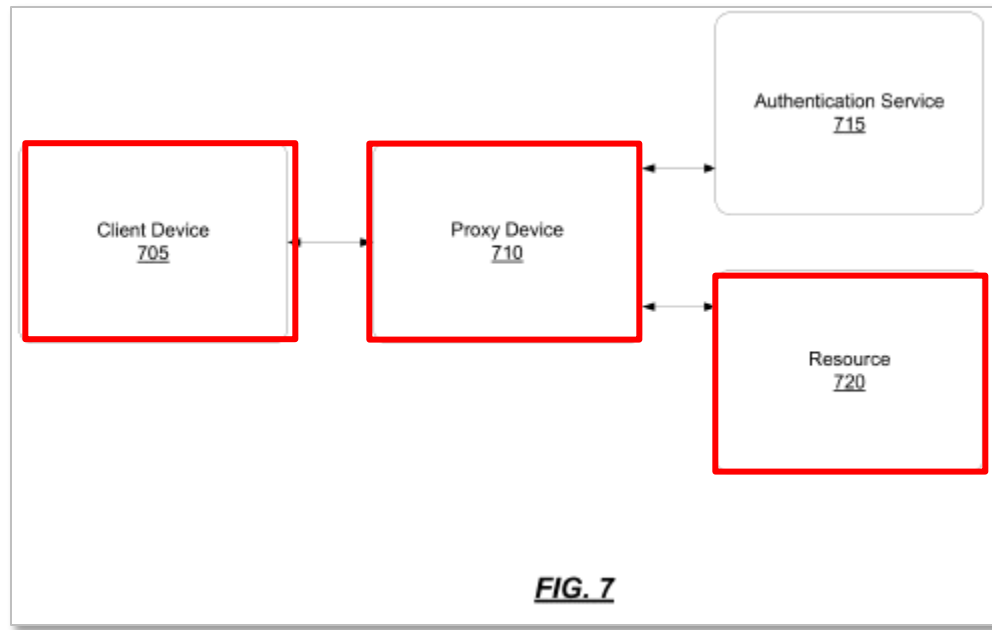
As illustrated in FIG. 5B [annotated below], the sandbox tool 110 can display a **user interface 535 for the first isolated execution environment . . .** in the desktop GUI 505. The user interface 535 can include various menus and widgets for accessing application programs, such as a tool bar 550 and application icon 555 for accessing **a web browser application program.**

EX1006, 12:22-29, FIG. 5B (annotated).



EX1006, FIG. 5B. (annotated).

As shown in Figure 7 below, Innes discloses “a proxy device 710” (“*proxy device*”) that provides a “client device 705” access to “resources 720, such as documents, emails, services, files, and the like.” EX1012, 22:13-15.



EX1012, FIG. 7 (annotated).

As explained above, a POSA would have been motivated, with a reasonable expectation of success, to implement the Walsh-AAPA user computing system on a network including Innes's proxy device and authentication services in order to securely access WebEx content on Walsh's network. EX1003, ¶182. Walsh discloses that computing devices may implement content isolation by using a "user execution environment 220" (for trusted content) and an "isolated execution environment 230" (for untrusted content). EX1006, FIG. 2. The computing devices access the trusted/untrusted content over a network using a web browser. EX1006, 3:64-4:4. WebEx is used within a web browser. EX1003, ¶182. The content, Innes teaches, can be "securely" accessed "using a proxy device." EX1012, 1:6-8. And Innes's "proxy device" allows computing devices to request certain authentication

services, which “may be provided by an authentication server.” EX1012, 27:43-61.

Moreover, this combination would have been obvious because it would amount to nothing more than combining familiar elements according to their known methods of use. *KSR*, 550 U.S. at 416.

In the Walsh-AAPA-Innes system, Walsh’s web browser would access WebEx (“*multi-user interactive software*”) to communicate with (“*send data to*”) an untrusted website (e.g., a WebEx server) (“*an untrusted destination*”) when Walsh’s “isolated execution environment 230” (“*isolated computing environment*”) has been “*authenticated*” by Innes’s authentication server (“*authentication device*”).

4. Claims 2 and 10

Claims 2 and 10 depend from claims 1 and 9, respectively, and recite “*wherein the multi-user interactive software application allows a remote machine to control an application or a process of the second set of one or more applications or processes running in the isolated computing environment, wherein controlling the application or process comprises at least one of viewing, interacting with, modifying, uploading data to, or downloading data from the application or the process.*” EX1003, ¶184. The Walsh-AAPA-Innes combination discloses this limitation.

These claims recite functionality that was available through WebEx—an application that was commercially available before the earliest priority date of the ’601 patent. As discussed for Element 1G, Walsh’s “web browser application[]” would access WebEx (“*a multi-user interactive software application*”). And a POSA would have understood that Walsh’s “web browser application” accessing WebEx enables “*viewing, interacting with, modifying, uploading,*” and “*downloading data.*” During a WebEx meeting, for example, a presenter can share their desktop, an application, or a document. EX1011, 0001-02; EX1003, ¶185.

5. Claims 3 and 11

Claims 3 and 11 depend from independent claims 1 and 9, respectively, and recite, “*wherein the multi-user interactive software application allows a remote device to control one or more of a pointer device, a keyboard input, or a display of the [system/isolated computing environment].*” The Walsh-AAPA-Innes combination teaches this limitation.

As noted above, the ’601 patent admits that WebEx discloses the “*multi-user interactive software application*” element. EX1001, 4:45-47. WebEx would be used on Walsh’s computer system, either within a browser or as a standalone application. EX1003, ¶¶186-187.

A POSA would understand that tools such as WebEx allow “*a remote device to control . . . a display of the system.*” EX1003, ¶¶188-189. During a WebEx

meeting, a presenter (“*a remote device*”) can share content with participants. For example, the presenter can share their desktop, an application, or a document.

EX1011, 0001-02.

While sharing, the presenter can control what the other participants see. For example, the presenter can synchronize views of a participant (“*control . . . a display of the system*”), so the participant’s view is matched to the presenter’s.

EX1011, 0001-02. A POSA would have understood that synchronizing the display of a participant controls the display of a system. EX1003, ¶190.

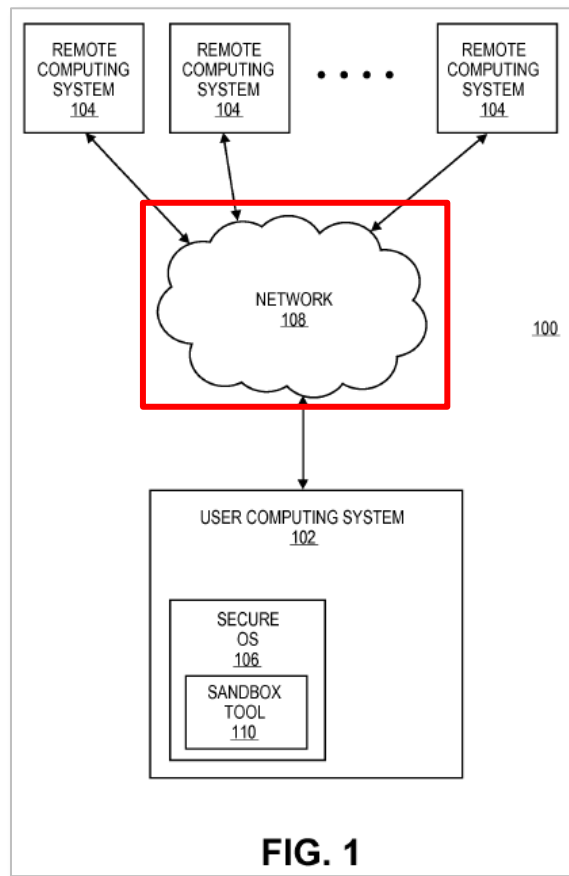
In the combined Walsh-AAPA-Innes system, Walsh’s web browser that runs a WebEx meeting (“*multi-user interactive software application*”) would enable a presenter (“*a remote device*”) to share and control content displayed on the client device (“*control . . . a display of the [system/isolated computing environment]*”).

6. Claims 4 and 12

Claims 4 and 12 depend from claims 1 and 9, respectively, and recite “*[wherein the system is connected/connecting] to a network, the network corresponding to one or more of a local area network (LAN), a wide area network (WAN), a cellular network, a Wi-Fi network, a digital subscribers line (DSL) network, a cloud computing network, a data center, or a satellite network.*”

EX1003, ¶¶192-193. Walsh discloses these claims.

Referring to Figure 1 (annotated below), Walsh discloses that “user computing system 102” is connected to “networks 108,” which can be “any type of communications networks, whether wired or wireless, to allow the computing system to communicate, such as **wide-area networks** (e.g. Internet) or **local-area networks**.” EX1006, 3:58-63.



EX1006, FIG. 1 (annotated).

7. Claims 7 and 15

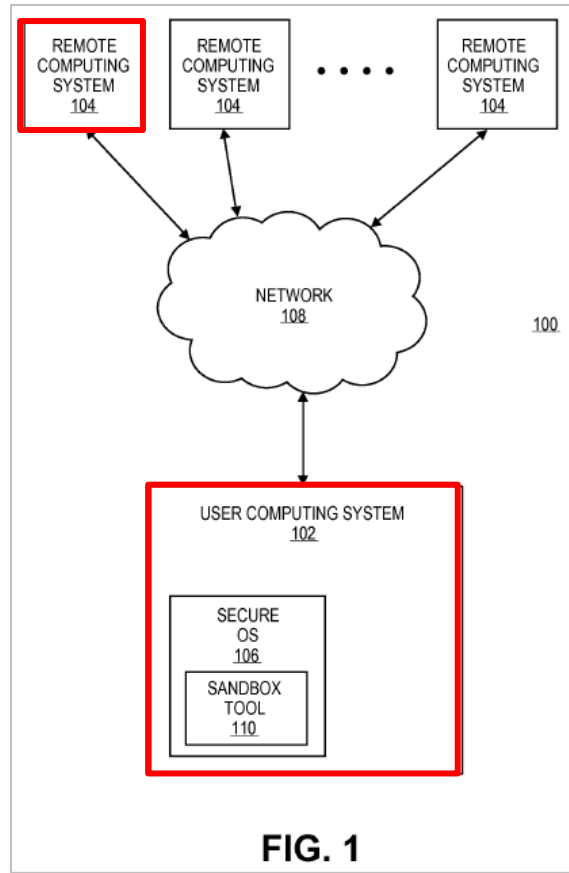
Claims 7 and 15 depend from claims 4 and 12, respectively, and recite
“*[wherein the processor is further configured to: run/running] one or more trusted*

applications in the first memory space, and wherein the one or more trusted applications communicate with one or more trusted machines via the network.”

EX1003, ¶¶194-195. Walsh discloses this limitation.

As described for Element 1C, Walsh discloses that a “user computing system 102” can execute “*trusted applications*” in “user execution environment 220,” which is supported by its own “user namespace” (“*first memory space*”).

EX1006, 5:32-40. The user execution environment 220 “allows a user to run applications programs” and “communicate with the remote computing systems 104” (“*trusted machines*”) via a network 108. EX1006, 11:58-64.



EX1006, FIG. 1 (annotated).

B. Ground 2: Claims 5, 6, 13, and 14 are obvious over Walsh, AAPA, Innes, and Netfilter.

- 1. A POSA would have been motivated to combine Walsh, AAPA, and Innes with Netfilter based on Walsh's disclosure.**

As described for Element 1J and claim 7, the Walsh-AAPA-Innes system allows “user computer system 102” to access information from “remote computers 104” over a “network 108.” And Walsh discloses that the “user computer system 102” may use “security enhanced Linux (‘SELinux’).” EX1006, 3:44-45. To the extent that neither Walsh nor Innes provide implementation details about firewalls

for Linux computers connected to the Internet, Netfilter does. Thus, a POSA would have been motivated to combine Walsh-AAPA-Innes with Netfilter to provide added security to the content isolation disclosed by Walsh. EX1003, ¶196.

Netfilter discloses a “packet filtering framework” for use with the Linux operating system. EX1013, 1. In particular, Netfilter allows users to “build internet firewalls based on stateless and stateful packet filtering.” EX1013, 1.

A POSA would have been motivated, with a reasonable expectation of success, to use Netfilter’s firewall within Walsh’s computer system because this combination would amount to nothing more than combining familiar elements according to their known methods of use. *KSR*, 550 U.S. at 416. Again, Walsh discusses protecting a computer system using the Linux operating system. And Netfilter discloses a firewall built into the Linux operating system. Thus, a POSA would have reasonably expected that Netfilter’s firewall would work in Walsh’s computer system to further protect it. EX1003, ¶198.

2. Claims 5 and 13

Claims 5 and 13 depend from claims 4 and 12, respectively, and recite “*[wherein the processor is further configured to: implement/implementing] a host-based firewall, wherein the host-based firewall isolates the [system/first memory space and the second memory space] from communicating with one or more other*

devices on the network.” The Walsh-AAPA-Innes-Netfilter combination discloses this element.

Again, Walsh discloses computers that implement the Linux operating system. EX1006, 3:44-45. And Netfilter discloses a Linux-based software framework for building “internet firewalls” (“*a host-based firewall*”). EX1013, 1. Netfilter’s firewall regulates incoming and outgoing traffic by “stateless and stateful packet filtering” (“*isolates the system from communicating with one or more other devices on the network*”). EX1013, 2.

In the Walsh-AAPA-Innes-Netfilter combined system, Walsh’s “processor” (“*the processor*”) would implement Netfilter’s “internet firewall[]” (“*host-based firewall*”) that regulates incoming and outgoing traffic (“*isolates the [system/first memory space and the second memory space] from communicating with one or more other devices on the network*”). EX1003, ¶¶199-201.

3. Claims 6 and 14

Claims 6 and 14 depend from claims 5 and 13, respectively, and recite “*wherein the host-based firewall blocks incoming communications from the network that are sent to the workspace.*” The Walsh-AAPA-Innes-Netfilter combination discloses this limitation.

Again, Walsh’s user execution environment (“*the workspace*”) allows a user to execute applications and programs. EX1006, 5:30-32. And Netfilter’s firewall

(“*the host-based firewall*”) regulates incoming traffic (“*blocking incoming communication from the network*”) based on a set of packet-filtering rules.

EX1013, 2.

In the Walsh-AAPA-Innes-Netfilter combined system, Walsh’s computing system would implement Netfilter’s firewall (“*host-based firewall*”) that regulates incoming and outgoing traffic (“*block incoming communication from the network*”) based on an application executing in the workspace (“*sent to the workspace*”).

EX1003, ¶¶202-204.

C. Ground 3: Claims 8 and 16 are obvious over Walsh, AAPA, Innes, and Franco.

1. A POSA would have been motivated to combine Walsh, AAPA, and Innes with Franco based on Walsh’s disclosure.

Again, Walsh discloses a “sandbox tool 110” (“*internal isolation firewall*”) that isolates the “isolated execution environment 230” (“*isolated computing environment*”) from the “user execution environment 220” (“*workspace*”).

EX1006, 5:52-6:3 In some instances, Walsh discloses that content from the user execution environment 220 can be transferred to the namespace for the isolated execution environment 230:

[S]andbox tool 110 can copy necessary content and content 225 to the namespace for the isolated execution environment 230. For example, the sandbox tool 110 can copy or can instruct the secure OS 106 to copy the necessary content and the content 225 to the home directory and

tmp directory in the same location it would be found in the home directory and tmp directory of the user execution environment 220. The necessary content can be any content (files, applications, programs) which is required by the isolated execution environment 230 to function similarly as the user execution environment 220. The content 225 can be application programs, files, data, etc. that the user specified in the request

EX1006, 10:11-22.

Walsh also discloses that the security context for the isolated execution environment 230 can be “provided by the user of the computing system 102.”

EX1006, 10:58-60. To the extent that Walsh does not provide details about such user-provided security contexts, Franco does. A POSA wishing to implement Walsh’s system thus would have been motivated to combine it with Franco.

EX1003, ¶207.

Like Walsh, Franco is directed to preventing malware or other malicious software from compromising a host computer. EX1009, 1:22-29. To address this issue, Franco executes applications with “reduced privilege” so that they’re restricted from accessing portions of the host. EX1009, 2:31-36. When attempts to access a restricted area of the host are made, Franco captures explicit user input to authorize the access. EX1009, 2:36-56.

A POSA would have been motivated, with a reasonable expectation of success, to use Franco’s user-prompt mechanism within Walsh’s user computing

system because this combination would amount to nothing more than combining familiar elements according to their known methods of use. *KSR*, 550 U.S. at 416. Both Walsh and Franco disclose isolating processes within a computing device. In particular, Walsh discloses a “sandbox tool” that creates “an isolated execution environment 230 for accessing content 225 (trusted or untrusted) without exposing other processes such as the user execution environment 220 and resources of the user computing system 102 to the content 225.” EX1006, 5:52:57. And Franco discloses prompting a user to facilitate communication between an application and a restricted area of the computing device. In the combined system, Walsh’s “sandbox tool” would require Franco’s user input when data is transferred between isolated and non-isolated processes. EX1003, ¶¶205-209.

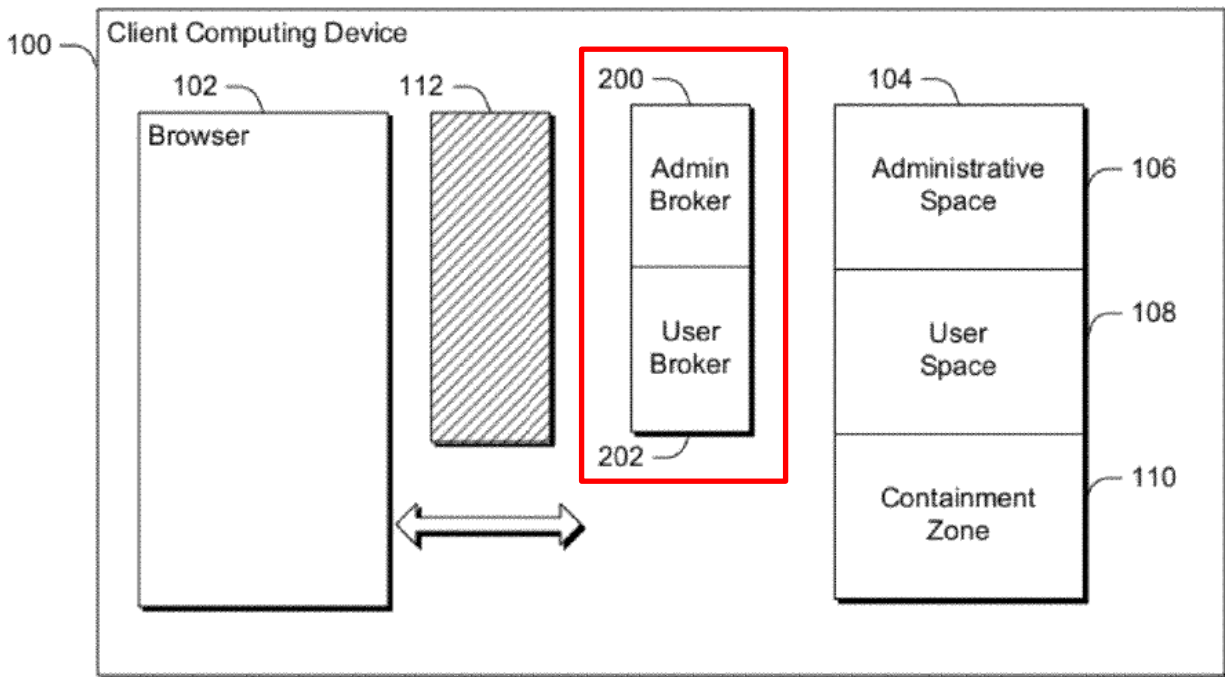
2. Claims 8 and 16

Claims 8 and 16 depend from claims 1 and 9, respectively, and recite “*wherein the internal isolation firewall prompts a user of the [system/workspace] to allow communication between the first memory space and the second memory space.*” EX1003, ¶¶210-213. The Walsh-AAPA-Innes-Franco combination discloses this limitation.

Again, Walsh discloses a “sandbox tool 110” (“*internal isolation firewall*”) that isolates the “isolated execution environment 230” (“*isolated computing environment*”) from the “user execution environment 220” (“*workspace*”).

EX1006, 5:52-57. And Walsh discloses that the security context for the isolated execution environment 230 can be “provided by the user of the computing system 102.” EX1006, 10:58-60.

Franco prevents applications from accessing restricted areas of a computer and requires explicit user input when access attempts are made. EX1009, 2:36-56. When the restricted area is accessed, a broker object prompts the user with a dialog box, requiring the user to input credentials (*“prompts a user of the [system/workspace]”*) to allow the access attempt (*“to allow communication”*). EX1009, 5:4-11.



EX1009, FIG. 2 (annotated).

In the Walsh-AAPA-Innes-Franco combined system, Walsh’s “sandbox tool 110” (*“internal isolation firewall”*) would display Franco’s dialog box (*“prompt a*

user of the [system/workspace]”) when content is transferred between the isolated and non-isolated applications (“to allow communication between the first memory space and the second memory space”).

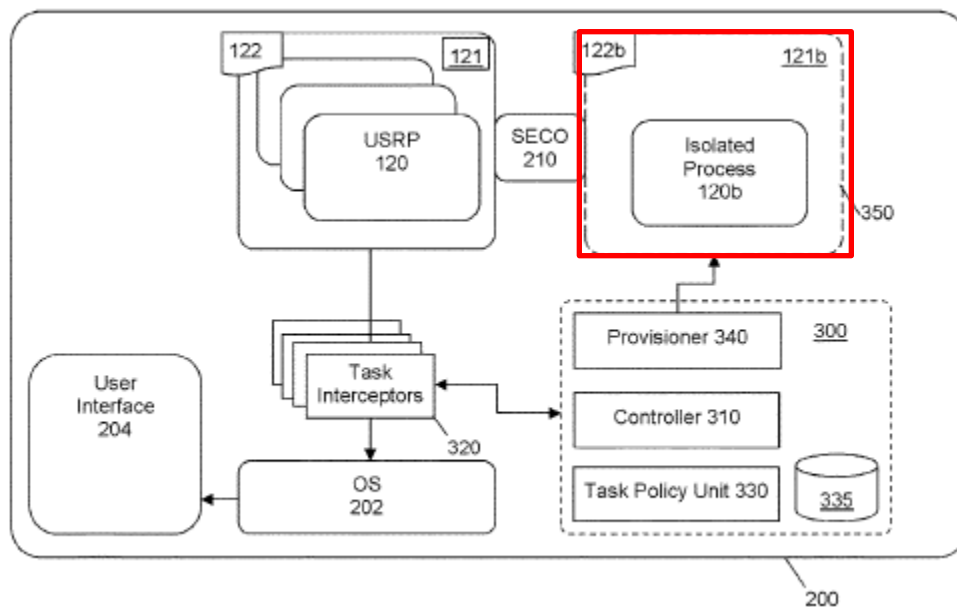
D. Ground 4: Claims 1-4, 7, 9-12, and 15 are obvious over Austin, AAPA, and Bloch.

1. A POSA would have been motivated to combine Austin with Bloch based on Austin’s disclosure.

Austin is directed to “isolating untrusted or malicious content on a clipboard.” EX1005, ¶3. Austin recognizes the “on-going need to protect computer devices from malicious content.” EX1005, ¶5. For example, Austin recognizes “untrusted content may be copied by the user from an untrusted source (e.g. untrusted content, untrusted application) . . .” EX1005, ¶15. Austin further recognizes risks posed by “untrusted tasks” received through “Web browsing, viewing email attachments, starting an untrusted application program, or accessing a particular file.” EX1005, ¶72.

To address these issues, Austin discloses content-isolation techniques to inhibit untrusted tasks “from interfering with other tasks or components of the computer device.” EX1005, ¶72. Isolation may involve “limiting the resources of the computer device which are accessible by the untrusted content.” EX1005, ¶15. As a result, the untrusted task and content may be manipulated. EX1005, ¶73.

Referring to Figure 4 (annotated below), Austin discloses a computer device 200 with an operating system 202 where “each process 120 that is initiated for a particular user will be run in a security context 121 that derives access rights and permissions from the user’s security account.” EX1005, ¶66. “For example, a particular process 120 is able to read from, but not write to, a file of the first resource 115.” EX1005, ¶68.



EX1005, FIG. 4 (annotated).

Austin handles untrusted tasks by opening them in a “task isolation environment or sandbox 350.” EX1005, ¶86. “Where the controller 310 determines that the task in question relates to untrusted content, i.e. is determined to be an untrusted task, then the controller 310 causes the untrusted task to be opened in a task isolation environment or sandbox 350.” EX1005, ¶86. A temporary user

account may be created for the sandbox, such that the isolated task uses the temporary user account for data and registry access. EX1005, ¶91. As a result, “[a]ny malicious activity only affects the temporary user account 121 b and does not affect the primary user account 121.” EX1005, ¶91.

In addition to isolating untrusted processes, Austin discloses another mechanism to protect the host from malware that may attempt to access the host system’s clipboard. EX1005, ¶13. Austin recognizes that “most conventional operating systems” would allow the isolated untrusted process “to access the primary clipboard,” introducing an additional vulnerability. EX1005, ¶102.

Austin addresses this issue via “a primary clipboard accessible from a primary user account” and a “secondary clipboard . . . provisioned and associated with the secondary user account” within the isolated sandbox. EX1005, ¶20. Each clipboard is supported by its own respective memory space. The primary clipboard 718 is supported by a primary clipboard memory 819 (i.e., “physical memory within the hardware 201”). EX1005, ¶119. “[T]he secondary clipboard 728 is supported by appropriate physical memory 829.” EX1005, ¶119.

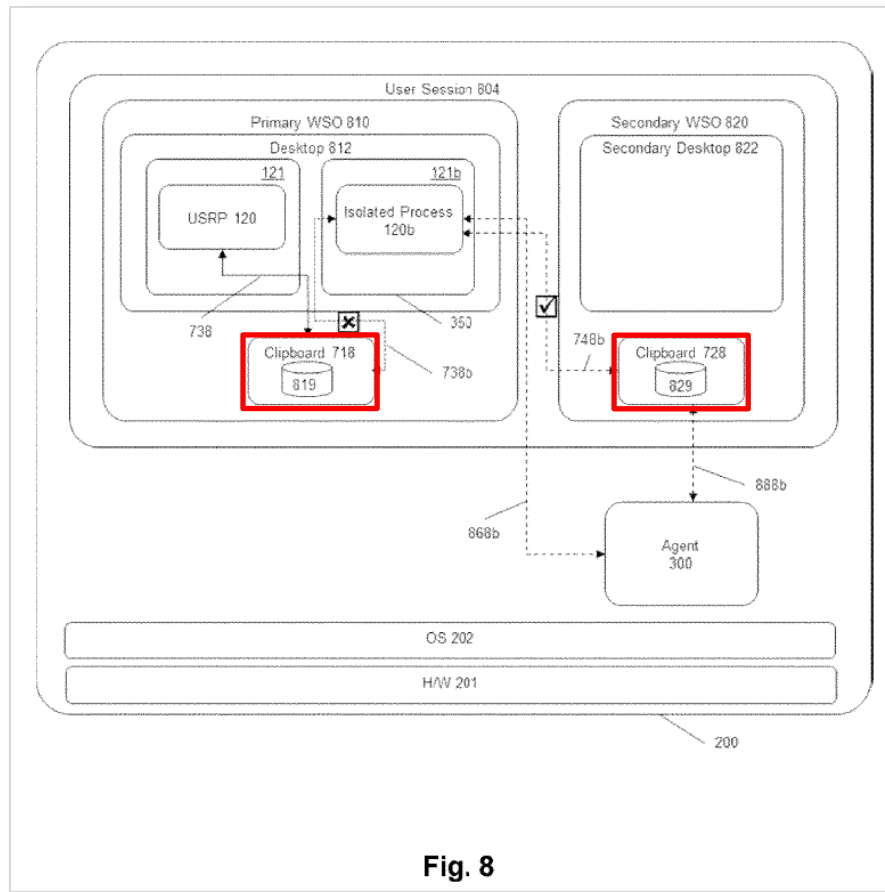


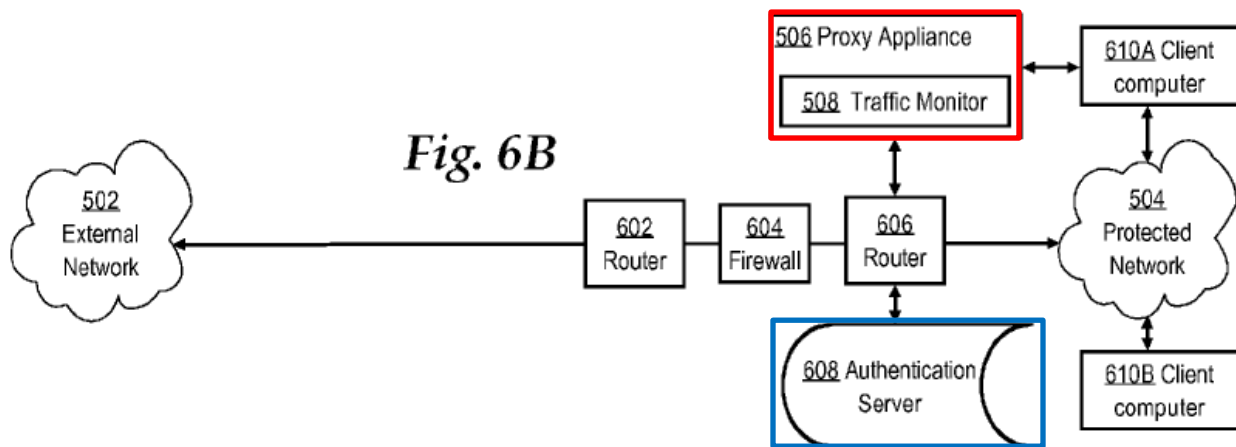
Fig. 8

EX1005, FIG. 8 (annotated).

A POSA would have been motivated to combine Austin with Bloch based on the teachings in Austin itself. Austin discloses mechanisms to “protect the computer device 200 from malicious attacks.” EX1005, ¶74. And because Austin discloses networked based threats in the form of untrusted webpages and untrusted email attachments, a POSA reading Austin would have sought additional references such as Bloch to implement additional protection mechanisms directed at network-based threats. EX1005, ¶72; EX1003, ¶220.

Bloch confronts a problem associated with network-based threats, in that administrators may wish to manage (i.e., allow, block) network traffic. EX1007, 6:51-53. Bloch addresses this problem by using a proxy device to inspect and filter out suspicious content. EX1007, 3:35-40. The proxy device employs firewall rules to filter traffic, uses signature-based filters to avoid malware and spyware, and blocks certain downloads. EX1007, 3:30-48.

Bloch discloses an “apparatus and computer program product for managing and monitoring network traffic and filtering responses.” EX1007, 1:2-4. Specifically, a proxy appliance manages “network traffic into and out of an internal network to block or redirect the traffic to avoid malware.” EX1007, 3:36-38. “The proxy appliance 506 is coupled to an authentication server 608 so that the proxy appliance can provide an integrated user authentication service, for example, using lightweight directory access protocol (LDAP), Microsoft Active Directory, etc.” EX1007, 8:20-24. The proxy appliance can also include “application proxies for hyper text transfer protocol (HTTP), hyper text transfer protocol secure (HTTPS), and file transfer protocol (FTP).” EX1007, 3:41-44. The proxy appliance uses “web reputations filters that analyze different web traffic and network-related parameters to evaluate the trustworthiness of a given URL.” EX1007, 3:55-57.



EX1007, FIG. 6B (annotated).

As an example operation, the proxy appliance may receive a “client request” for “information from an HTTP server, FTP server, or other network resource that is located outside the protected network.” EX1007, 5:21-25. The system checks whether a user associated with the computer device has been authenticated. EX1007, 5:25-28. If the user is authenticated, “then several checks are made for parameters associated with the client request.” EX1007, 5:35-37. For example, the client and server IP addresses, as well as the requested URL, are checked against both blacklists and whitelists. EX1007, 5:43-54. Additional checks, such as determining whether a user-agent and requested file extension are within a blacklist, are performed. EX1007, 5:60-66. Based on the checks, the content may be fetched from the server and sent to the client device. EX1007, 5:45-50.

A POSA would have been motivated to combine Austin with Bloch to “authenticate the isolated computing environment” and “send data to an untrusted destination . . . via a proxy device” because the combination would further protect Austin’s computing device. EX1003, ¶224. As stated above, Austin recognizes the threat posed by untrusted tasks such as accessing untrusted websites. EX1005, ¶73. To protect the computing device, Austin isolates untrusted tasks to prevent interference with other tasks and computing device components. EX1005, ¶72. As a further layer of protection, a POSA would have been motivated to use Bloch’s proxy appliance and authentication server to access an untrusted website (“*an untrusted destination*”). EX1003, ¶224.

To do so, a POSA would first authenticate Austin’s isolated environment with Bloch’s proxy appliance to establish a layer of trust. EX1003, ¶225. As discussed, Austin is concerned with isolating untrusted tasks, whether that involves accessing untrusted content or communicating with untrusted network entities (e.g., an untrusted website). EX1005, ¶¶72-73. To confirm that Bloch’s proxy device can be trusted, a POSA would have been motivated to perform authentication between the Austin’s isolation environment and Bloch’s proxy device via Bloch’s authentication server. EX1003, ¶225. Once authenticated, Bloch’s proxy device would be used to further protect Austin’s computing device. *Id.*

Bloch's proxy appliance would block or redirect network traffic based on domains and/or IP addresses included within the traffic. EX1007, 6:4-22; EX1007, 10:18-29. For example, Bloch's proxy appliance blocks traffic from a domain that is known to distribute spyware or other malware. EX1007, 7:52-61. When combined with Austin's isolation environment, Bloch's proxy appliance would be used to block traffic from sources confirmed to distribute malware or spyware, further protecting Austin's computing device. EX1003, ¶226.

In the Austin-Bloch combination, Austin's device would be able to access network content via Bloch's proxy appliance and authentication server. EX1003, ¶127. This combination amounts to nothing more than using a known technique according to known methods of use. *Id.* Connecting Bloch's proxy device and authentication server with Austin's device was a known technique to improve similar devices, methods, or products in the same way (e.g., to use a proxy device to protect computer functionality that is connected to a network). *Id.* For example, Austin discloses accessing content from websites. EX1005, ¶¶72-73. Bloch's proxy appliance is used to provide network resources (e.g., HTTP resources) to a computing device. EX1007, 4:21-50. Thus, Austin's computing device would access network content via Bloch's proxy appliance. EX1003, ¶127.

A POSA would also have a reasonable expectation of success authenticating Austin's isolated environment with Bloch's authentication server. EX1003, ¶228.

Austin discloses authenticating a user attempting to log into the computer by validating a set of user credentials (e.g., username and password). EX1005, ¶62. Austin states that the credentials are validated locally or remotely, such as by an active directory domain controller. EX1005, ¶62. Thus, Austin already contemplates performing authentication with a remote entity and could therefore authenticate with Bloch's authentication server. EX1003, ¶228.

Moreover, a POSA would have a reasonable expectation of success authenticating Austin's isolated environment with Bloch's authentication server because the isolated environment is provided its own set of credentials. EX1003, ¶229. When "controller 310" creates a new "task isolation environment 350," it also creates a new, temporary user account. EX1005, ¶87. The new user account is assigned credentials used to log into the new account. EX1005, ¶87. Therefore, a user could provide the credentials of the new user account associated with the isolated environment to Bloch's authentication server. Once authenticated, the user would have access to the new user account and the associated task isolation environment. EX1003, ¶229.

2. A POSA would have been motivated to combine Austin-Bloch with AAPA

The '601 patent admits that the claimed "multi-user interactive software application" can be implemented via commercially-available products, such as WebEx. EX1001, 4:45-46. WebEx was a well-known product and was used as a

standalone application, an email client plugin, or within a web browser. EX1003, ¶¶230-231.

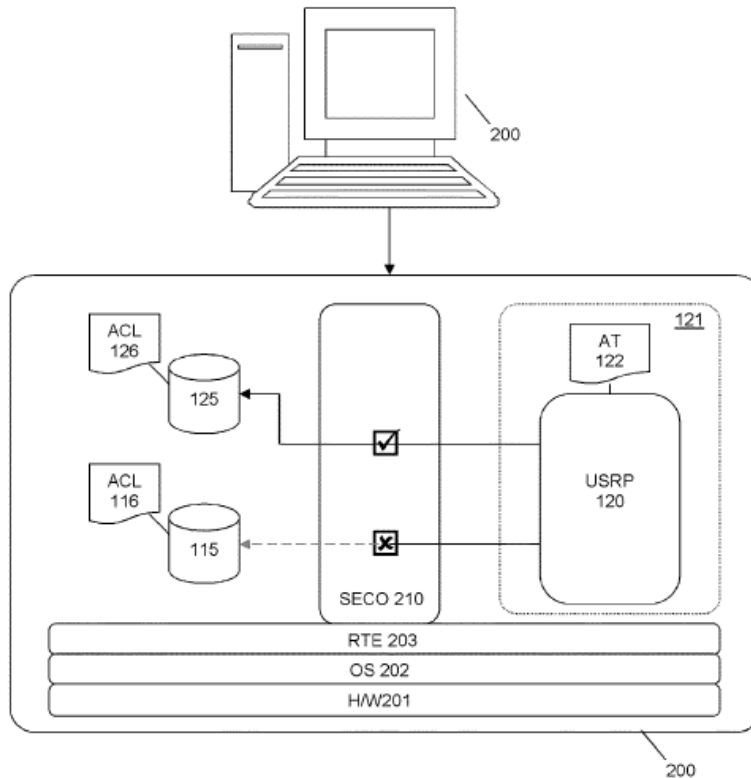
A POSA would have been motivated, with a reasonable expectation of success, to use Bloch's proxy appliance with Austin's client computer in order to securely access WebEx traffic via Austin's client computer. EX1003, ¶231. Austin explains that a client computer may access content over the Internet via a web browser application. EX1005, ¶72. A POSA would have understood that WebEx was a well-known application for accessing traffic over the Internet, and a POSA would have understood that WebEx would be (i) installed as a standalone program on Austin's client computer or (ii) used within a web browser or email client on Austin's client computer. EX1003, ¶231. The combined Austin-AAPA system would then be used on a network including Bloch's proxy appliance to provide Austin's client computer with secure access to network resources (e.g., an HTTP server). *Id.* Thus, it would have been obvious to make the Austin-AAPA-Bloch combination because it would have amounted to nothing more than combining familiar elements according to their known methods of use. *Id.*; *KSR*, 550 U.S. at 416.

3. Independent claims 1 and 9

(a) [1-pre] “A system” and [9-pre] “A method”

To the extent the preambles are limiting, Austin discloses them. EX1003, ¶¶232-233.

Austin discloses a computer device, method, and system. EX1005, Abstract, ¶5. Referring to Austin’s Figure 1 (reproduced below), “computer device 200 includes physical hardware (HW) 201 such as memory, processors (CPUs), I/O interfaces, backbone, power supply and so on.” EX1005, ¶63. Austin’s “computer device” implements a “*method*” because it creates a “task isolation environment” to run untrusted content in order to limit “the resources of the computer device which are accessible by the untrusted content.” EX1005, ¶15.

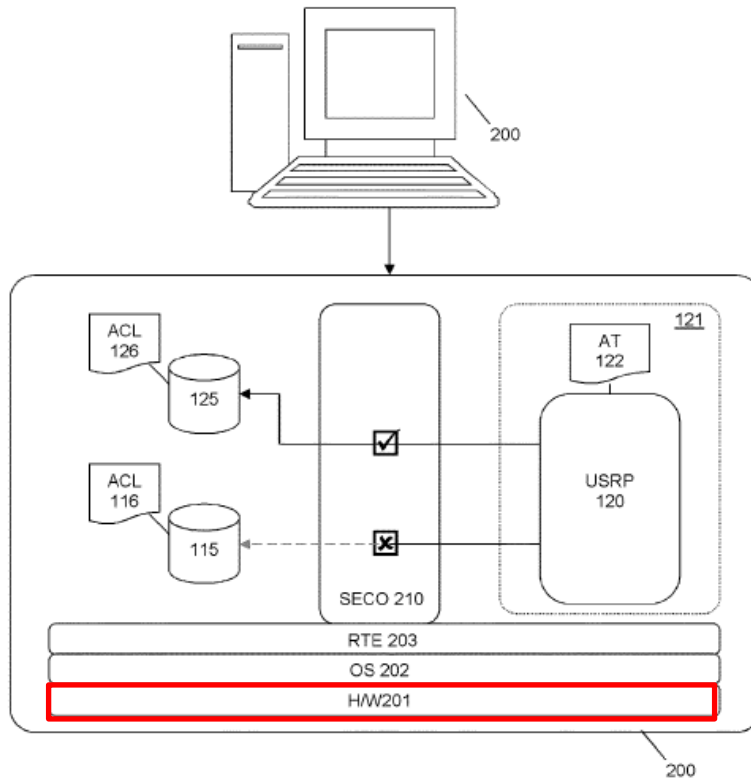


EX1005, FIG. 1.

(b) [1A] “a memory”

Austin discloses “a memory.” EX1003, ¶¶234-235.

Austin’s FIG. 1, shown below, depicts “a schematic overview of the computer device 200.” EX1005, ¶62. “[C]omputer device 200 includes “physical hardware (HW) 201 such as *memory*. . . .” EX1005, ¶63 (emphasis added).

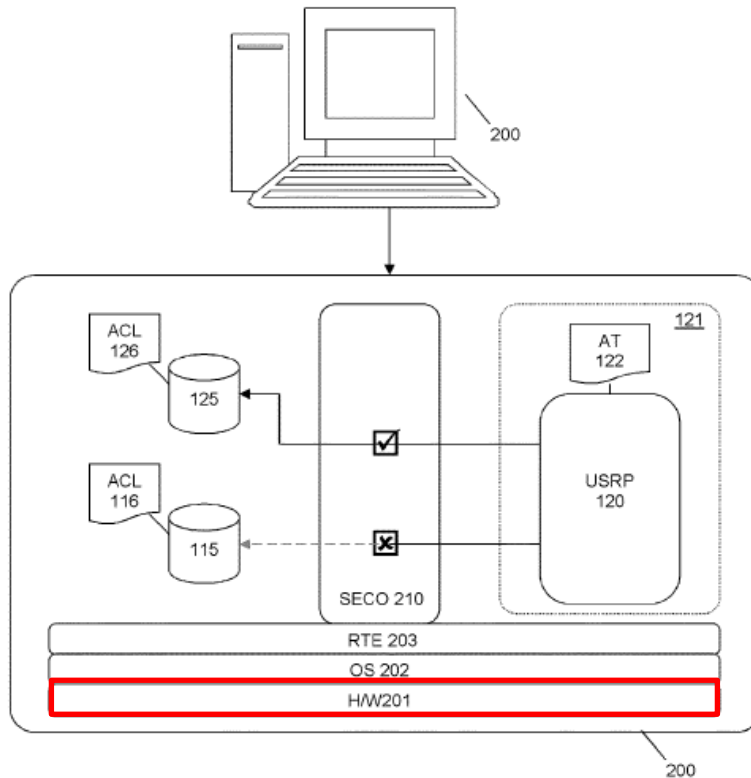


Austin, FIG. 1 (annotated).

(c) [1B] “a processor configured to”

Austin discloses “*a processor.*” EX1003, ¶236.

Referring again to FIG. 1 (shown below), Austin’s “computer device 200 includes physical hardware (HW) 201 such as memory, *processors (CPUs).* . . .” EX1005, ¶63 (emphasis added).



EX1005, FIG. 1 (annotated).

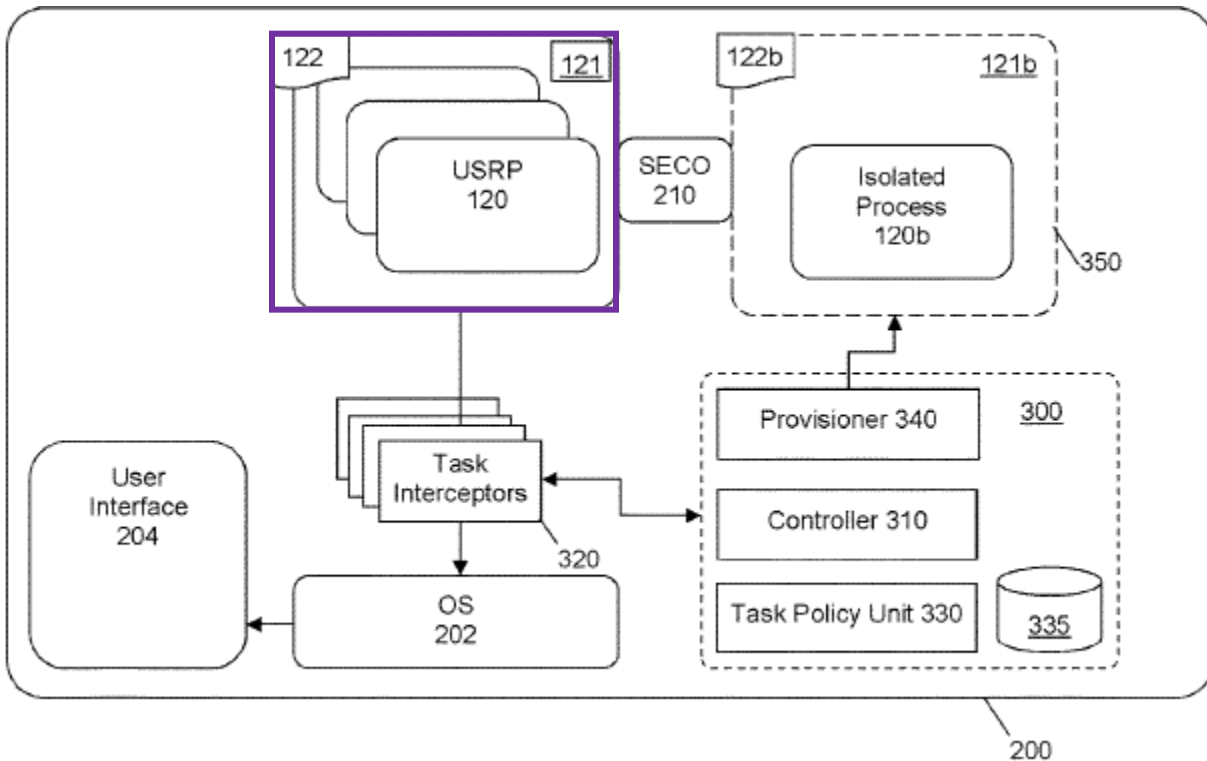
- (d) [1C] and [9A] “[implement/implementing] a workspace that uses a host operating system, wherein the workspace enables operation of a first set of one or more applications or processes via a first memory space”

Austin discloses this element. EX1003, ¶¶237-243.

Again, Austin’s “computer device 200” includes “physical hardware (HW) 201,” with a processor and memory to support “operating system (OS) 202.”

EX1005, ¶63. The operating system provides “user account 121” (“a workspace that uses a host operating system”). EX1005, ¶87; *see also* EX1005, ¶68 (OS provides resources to user account processes). And Austin’s user account 121 (in

purple below) supports execution “of one or more user processes 120” (“*enables operation of a first set of one or more applications or processes via a first memory space*”). EX1005, ¶70.



EX1005, FIG. 4 (annotated).

Austin’s “user account” has its own memory and “namespace” (“*a first memory space*”). EX1005, ¶95. Austin discloses a “primary clipboard . . . accessible from a primary user account” EX1005, ¶20. The primary clipboard 718 is supported by a primary clipboard memory 819 (i.e., “physical memory within the hardware 201”). EX1005, ¶119, FIGs. 7 (shows applications in the memory space), 8 (shows physical first memory space). A POSA would have understood

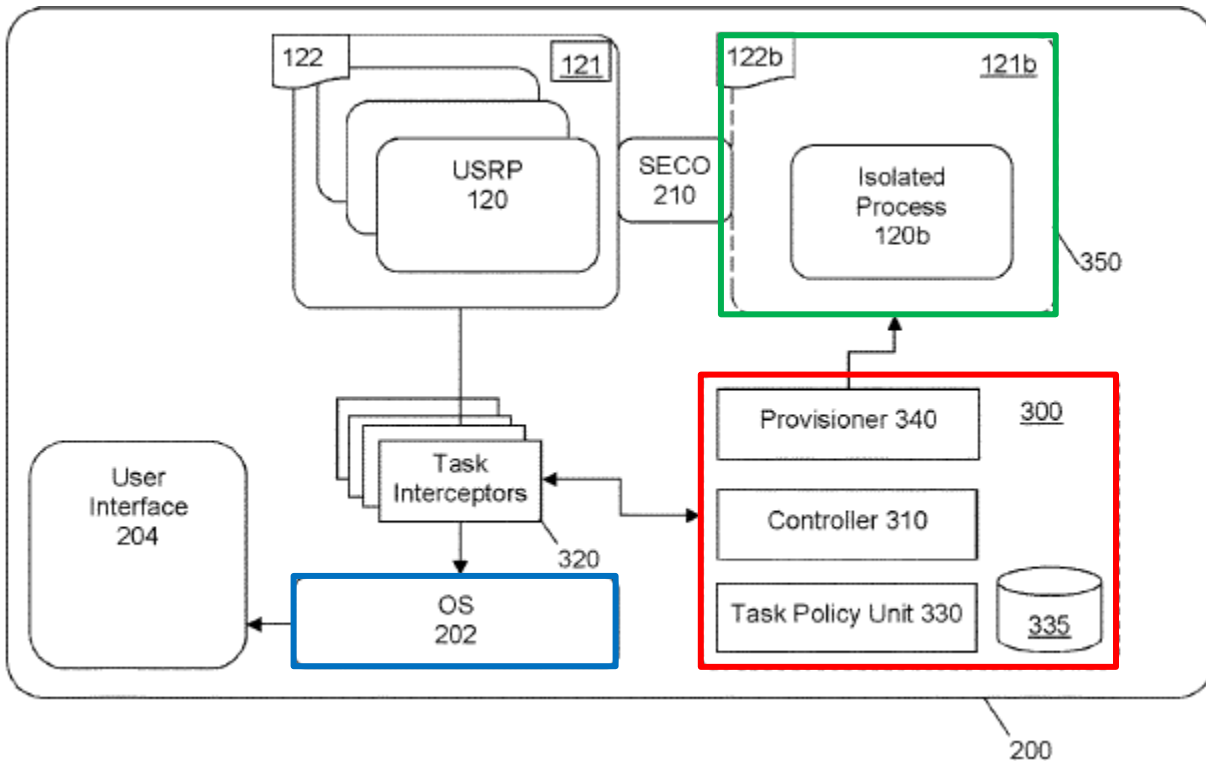
that Austin’s user account encompasses resources, including its own memory, available to the user account. EX1003, ¶¶241-243.

(e) [1D] and [9B] “[implement/implementing] an isolated computing environment that uses the host operating system”

Austin discloses this element. EX1003, ¶¶244-247.

Austin discloses an “isolation environment 350” (“*an isolated computing environment*”) that is implemented by “content isolation agent 300.” EX1005, ¶74. Referring to Austin’s Figure 4 (annotated below), content isolation agent 300 (in red) includes “a controller unit 310” that causes untrusted tasks “to be opened in a task isolation environment or sandbox 350” (in green). EX1005, ¶¶74, 86.

Austin’s “isolation environment 350” (“*isolated computing environment*”), which is implemented by content isolation agent 300, “*uses the host operating system,*” as claimed, because Austin discloses that the content isolation agent 300 either “operates in close association with the native operating system 202 [blue below]” or is “integrated within the operating system 202.” EX1005, ¶74.



EX1005, FIG. 4 (annotated).

- (f) [1E] and [9C] “wherein the isolated computing environment enables operation of a second set of one or more applications or processes via a second memory space”

Austin discloses this element. EX1003, ¶¶248-250.

Austin’s “isolation environment 350” (“*isolated computing environment*”) enables untrusted tasks to be executed (“*enables operation of a second set of one or more applications or processes*”). Again, the controller unit 310 causes untrusted tasks “to be opened in a task isolation environment or sandbox 350.” EX1005, ¶86. The untrusted tasks may include “non-executable content (data)” or “executable content (code).” EX1005, ¶72.

Austin's untrusted task ("*a second set of one or more applications or processes*") execute in the namespace of a "temporary user account" ("*a second memory space*"). EX1003, ¶249. According to Austin, the untrusted task "runs in this temporary user account 121 b and accesses the profile of the temporary user for both data access and registry access." EX1005, ¶91. The untrusted task "does not have access to the namespace of the primary user account 121." EX1005, ¶95. As a result, "[a]ny malicious activity only affects the temporary user account 121 b and does not affect the primary user account 121." EX1005, ¶91. Austin also discloses a "secondary clipboard . . . provisioned and associated with the secondary user account" within the isolated sandbox. EX1005, ¶20. "[T]he secondary clipboard 728 is supported by appropriate physical memory 829." EX1005, ¶119, FIGs. 7 (shows second set of applications), 8 (shows second memory space).

(g) [1F] and [9D] "wherein the second set of one or more applications or processes comprises a multi-user interactive software application"

Austin in view of AAPA discloses this element. EX1003, ¶¶251-254.

Again, Austin's untrusted tasks are the "*second set of one or more applications or processes*." These untrusted tasks include executable code, and applications such as web browsers and email clients. EX1005, ¶¶72-73, 81. The '601 patent admits that the claimed "*multi-user interactive software application*" existed in commercially-available products, such as WebEx. EX1001, 4:45-46.

And a POSA would have understood that WebEx could be used (i) within a web browser, (ii) as a standalone application, or (iii) as an email client plugin. EX1003, ¶253.

A POSA would have been motivated, with a reasonable expectation of success, to execute WebEx (“*multi-user interactive software application*”) as one of Austin’s untrusted tasks so that any “malicious activity” on WebEx would affect only “the temporary user account 121 b and” would “not affect the primary user account 121.” EX1005, ¶91; EX1003, ¶254. This combination would have been obvious because it would have amounted to nothing more than combining familiar elements (e.g., Austin’s untrusted tasks and WebEx) according to their known methods of use. *KSR*, 550 U.S. at 416.

In the Austin-AAPA combined system, Austin’s untrusted task (“*the second set of one or more applications or processes*”) would be WebEx (“*a multi-user interactive software application*”).

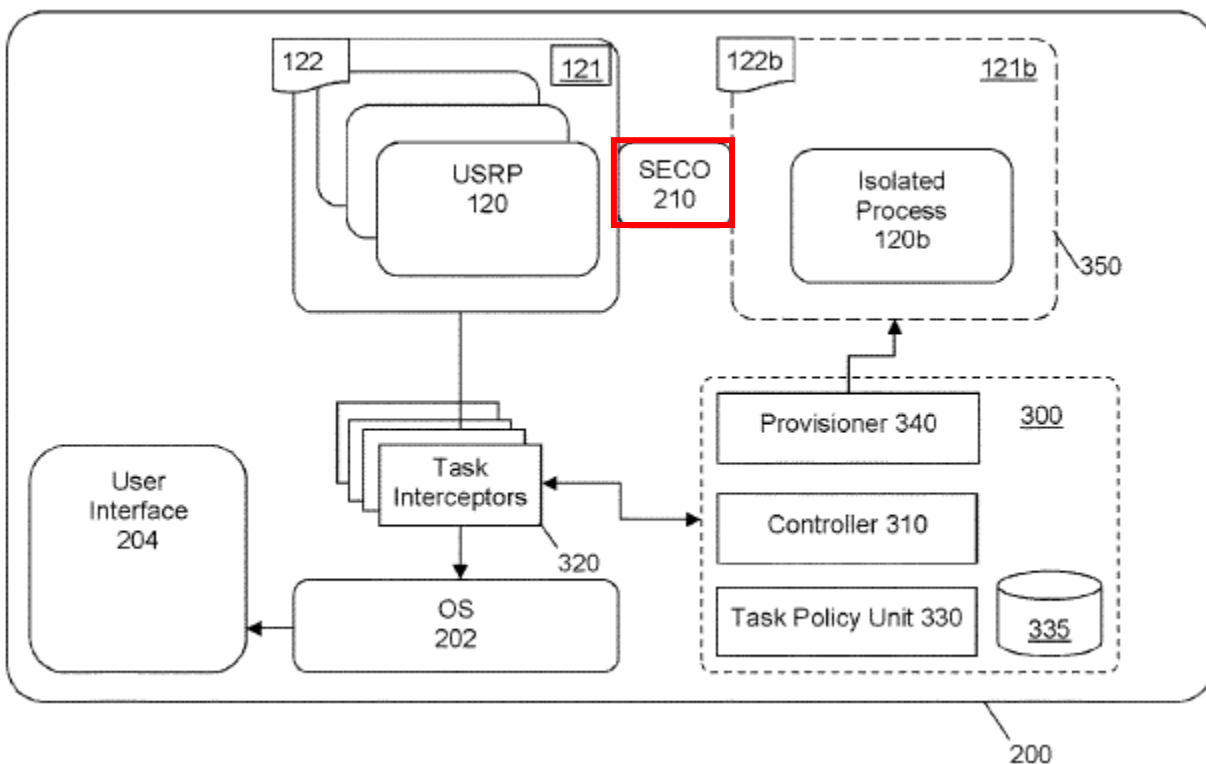
(h) [1G] and [9E] “[isolate/isolating] the isolated computing environment from the workspace using an internal isolation firewall”

Austin discloses this element. EX1003, ¶¶255-258.

An “internal isolation firewall,” the ’601 patent discloses, enforces “the segregation of the first and second memory spaces.” EX1001, 8:37-38. Like this

disclosure, Austin’s “security module 210” (“*internal isolation firewall*”) isolates an untrusted task “from content of the main user account 121.” EX1005, ¶89.

“[S]ecurity module 210 is arranged to perform an access check” when an isolated process requests access to a resource, such as “software, system services, drivers, files and/or registry settings.” EX1005, ¶¶63, 67. As a result, the untrusted task is unable to access data or the profile of user account 121. EX1005, ¶90. Additionally, “the untrusted task” is restricted to “temporary user account 121b for data and registry access.” EX1005, ¶91. Thus, “[a]ny malicious activity only affects the temporary user account 121 b and does not affect the primary user account 121.” *Id.*



EX1005, FIG. 4 (annotated).

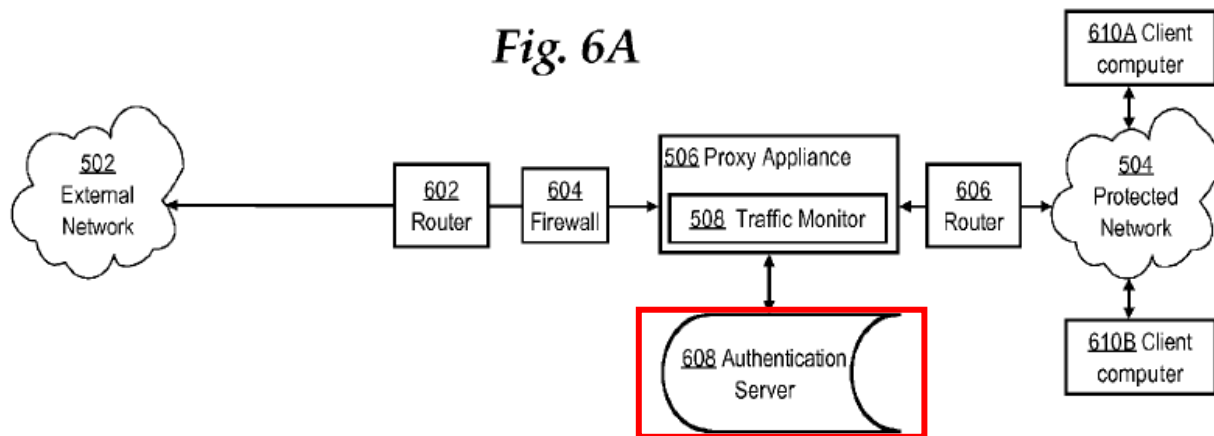
(i) [1H] and [9F] “[authenticate/authenticating] the isolated computing environment with an authentication device”

The Austin-AAPA-Bloch combination discloses this element. EX1003, ¶¶259-264.

Austin recognizes many operating systems tie access privileges to user accounts. EX1005, ¶60. As noted above, a new user account 121b and credentials may be created and associated with an isolation environment. EX1005, ¶87. The new user account within the isolation environment may be accessed by validating the credentials. EX1005, ¶87. This validation may occur at a local authentication system or via a remote authentication service. EX1005, ¶62. Austin discloses that the remote authentication service may be an “[a]ctive [d]irectory domain controller.” EX1005, ¶62. To the extent Austin does not provide implementation details about such a remote authentication service, a POSA interested in implementing the remote authentication service within Austin’s system would have been motivated to search for those details. That would have led to Bloch. EX1003, ¶260.

Bloch discloses an “authentication server 608 [red below]” to provide “integrated user authentication service, for example, using lightweight directory access protocol (LDAP), Microsoft Active Directory, etc.” EX1007, 8:22-24. For

example, when a user requests a resource, such as from an HTTP server, the user may first be authenticated.” EX1007, 5:25-28.



EX1007, FIG. 6A (annotated).

As explained above, a POSA would have been motivated to implement Austin’s computing device on a network including Bloch’s proxy appliance to provide further protection and securely access network content. To do so, the isolated task environment (“*the isolated computing environment*”) would authenticate with Bloch’s authentication server (“*an authentication device*”).

A POSA would have had a reasonable expectation of success using Austin’s computer device on a network including Bloch’s authentication server because it would amount to nothing more than combining familiar elements according to their known methods of use. *KSR*, 550 U.S. at 416; EX1003, ¶263. Austin discloses communicating over a network via the computer device by, for example, accessing a website via a web browser and accessing attachments received via email.

EX1005, ¶8. Bloch discloses client computers “coupled to protected network 504.”

EX1007, 8:26-27. Austin further discloses remotely authenticating user credentials, and assigning user credentials to the isolated task environment. EX1005, ¶¶62, 87. Bloch discloses authenticating user credentials. EX1007, 8:20-24.

In the Austin-Bloch combined system, Austin's "isolation environment 350" (*"isolated computing environment"*) would be authenticated via Bloch's "authentication server 608" (*"authentication device"*).

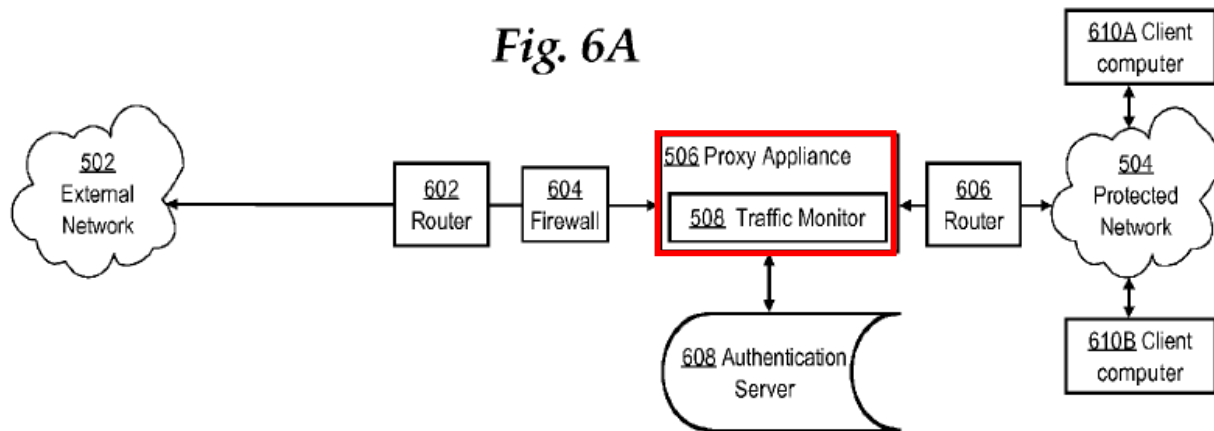
- (j) [1I] and [9G] "[send/sending] data to an untrusted destination from the multi-user interactive software application via a proxy device when the isolated computing environment has been authenticated."**

The Austin-AAPA-Bloch combination disclose this element. EX1003, ¶¶265-267.

Austin recognizes that untrusted tasks (i.e., tasks to isolate) may include forms of web browsing or executing untrusted applications. EX1005, ¶72. For example, a web browser may be isolated when instructed to access (*"send data to"*) an "untrusted website" (*"an untrusted destination"*). EX1005, ¶73 ("A trusted Web browser (e.g., Internet Explorer) may be instructed by the user to view an untrusted website."). WebEx may be used within a web browser. EX1003, ¶265.

In the Austin-AAPA-Bloch system, traffic from Austin's computer device (i.e., WebEx traffic) would pass through Bloch's proxy appliance 506. EX1007, 7:66-8:8, 8:39-44. Bloch's proxy appliance is configured to inspect and selectively

allow network traffic between client devices and, for example, an HTTP server (i.e., a website). EX1007, 5:22-24, 5:36-37, 5:49-50.



EX1007, FIG. 6A (annotated).

A POSA would have been motivated, with a reasonable expectation of success, to implement Austin-AAPA computer device on a network including Bloch's proxy appliance in order to securely access WebEx content on Austin's computing device. EX1003, ¶267. Austin discloses communicating over a network via the computer device by, for example, accessing a website via a web browser and accessing attachments received via email. EX1005, ¶8. WebEx is used within a web browser. EX1003, ¶267. Similarly, Bloch discloses client computers "coupled to protected network 504." EX1007, 8:26-27. Bloch further discloses a proxy appliance configured to receive and inspect "all network traffic." EX1007, 8:5-8. This combination would have been obvious because it would amount to nothing

more than combining familiar elements according to their known methods of use.

KSR, 550 U.S. at 416.

In the Austin-AAPA-Bloch combined system, an isolated web browser would access (“*send data to*”) an untrusted website (“*an untrusted destination*”), such as WebEx (“*a multi-user interactive software application*”), via Bloch’s “proxy appliance 506” (“*proxy device*”).

4. Claims 2 and 10

Claims 2 and 10 depend from claims 1 and 9, respectively, and recite “*wherein the multi-user interactive software application allows a remote machine to control an application or a process of the second set of one or more applications or processes running in the isolated computing environment, wherein controlling the application or process comprises at least one of viewing, interacting with, modifying, uploading data to, or downloading data from the application or the process.*” The Austin-AAPA-Bloch combination discloses this element. EX1003, ¶¶269-272.

These claims recite functionality that was available through WebEx—an application that was commercially available before the earliest priority date of the ’601 patent. As discussed for Elements [1I]/[9G], Austin’s “Web browser” would access WebEx (“*a multi-user interactive software application*”) as one of the untrusted tasks. And a POSA would have understood that Austin’s Web browser

accessing WebEx would enable “*viewing, interacting with, modifying, uploading data to, or downloading data from the application or the process.*” EX1003, ¶272.

During a WebEx meeting, for example, a presenter can share their desktop, an application, or a document. EX1011, 0001-02; EX1003, ¶272.

5. Claims 3 and 11

Claims 3 and 11 depend from claims 1 and 9, respectively, and recite “*wherein the multi-user interactive software application allows a remote device to control one or more of a pointer device, a keyboard input, or a display of the system.*” The Austin-AAPA-Bloch combination discloses this limitation. EX1003, ¶273.

As noted above, the ’601 patent admits that WebEx discloses the “*multi-user interactive software application*” element. EX1001, 4:45-47. And WebEx would be used on Austin’s client computer, either within a browser or as a standalone application. EX1003, ¶¶274-275.

A POSA would understand that tools, such as WebEx, allow “*a remote device to control . . . a display of the system.*” EX1003, ¶276. During a WebEx meeting, a presenter (“*a remote device*”) can share content with participants. For example, the presenter can share their desktop, an application, or a document. EX1011, 0001-02.

While sharing, the presenter can control what the other participants see. For example, the presenter can synchronize views of a participant (“*control . . . a display of the system*”), so the participant’s view is matched to the presenter’s. EX1011, 0001-02. A POSA would have understood that synchronizing the display of a participant controls the display of a system. EX1003, ¶¶277-278.

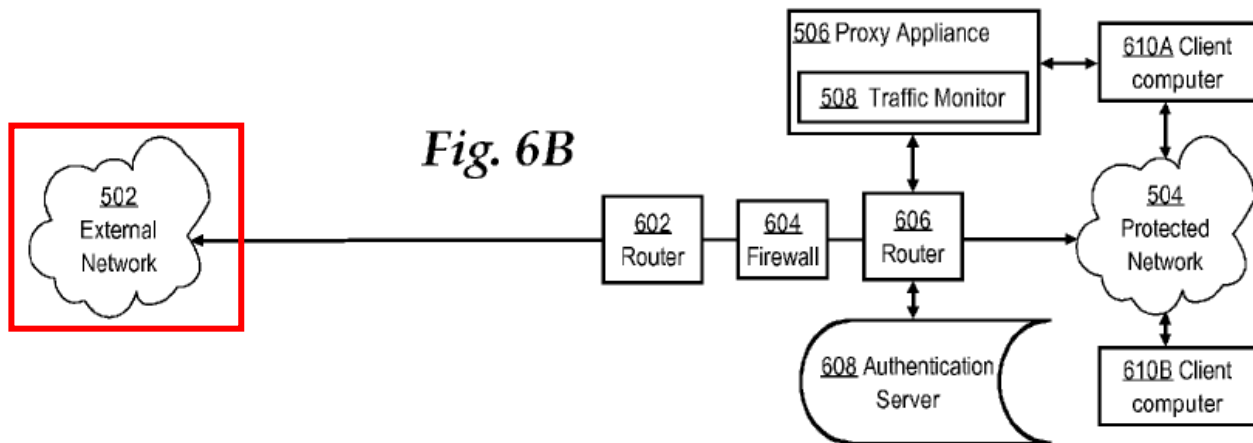
In the combined Austin-AAPA-Bloch system, Austin’s web browser that runs a WebEx meeting (“*multi-user interactive software application*”) would enable a presenter (“*a remote device*”) to share and control content displayed on the client device (“*control . . . a display of the [system/isolated computing environment]*”). EX1003, ¶279.

6. Claims 4 and 12

Claims 4 and 12 depend from claims 1 and 9, respectively, and recite “*[wherein the system is connected/connecting] to a network, the network corresponding to one or more of a local area network (LAN), a wide area network (WAN), a cellular network, a Wi-Fi network, a digital subscribers line (DSL) network, a cloud computing network, a data center, or a satellite network.*” The Austin-AAPA-Bloch combination discloses this limitation.

Again, Austin discloses that “untrusted tasks may include certain forms of Web browsing.” EX1005, ¶72. Bloch discloses an external network that may

include the internet. EX1007, 8:1-3. A POSA would have understood that the internet is a WAN, comprising a collection of LANs. EX1003, ¶¶280-281.



EX1007, FIG. 6B (annotated).

7. Claims 7 and 15

Claims 7 and 15 depend from claims 4 and 12, respectively, and recite
“[wherein the processor is further configured to: run/running] one or more trusted applications in the first memory space, and wherein the one or more trusted applications communicate with one or more trusted machines via the network.”

The Austin-AAPA-Bloch combination discloses this limitation. EX1003, ¶¶282-285.

As noted above, Austin executes a plurality of processes such as a trusted web browser (*“[run/running] one or more trusted applications”*). EX1005, ¶73. These processes execute under “primary user account 121.” EX1005, ¶¶102, 119.

Austin also recognizes that tasks related to trusted content may function without intervention (i.e., isolation). EX1005, ¶78. For example, a web browser may be used to access a webpage. EX1005, ¶73. Bloch discloses “detection techniques” for when a client is “seeking to retrieve information from an HTTP server” (i.e., web browsing). EX1007, 5:19-24. If the IP address of the requested server is in a whitelist (“*one or more trusted machines*”), the request is allowed and the server’s response is provided to the client (“*communicate . . . via a network*”). EX1007, 5:46-50.

Thus, in the combined Austin-AAPA-Bloch system, Austin’s computing device would execute a browser (“*[run/running] one or more trusted applications*”) under the namespace of the primary user account (“*in the first memory space*”) to retrieve content from a webpage (“*communicate . . . via the network*”) hosted by an HTTP server (“*a trusted machine*”).

E. Ground 5: Claims 5, 6, 13, and 14 are obvious over Austin, AAPA, Bloch, and Walker.

1. A POSA would have been motivated to combine Austin, AAPA, and Bloch with Walker based on Austin’s disclosure.

Austin discloses embodiments that use the Windows operating system. EX1005, ¶64. For example, Austin states that the security module may be “a security sub-system or security manager” that enacts a Windows security model. EX1005, ¶65. To the extent that Austin does not describe implementation details or

functionality of the Windows security model, a POSA would have been motivated to combine the Austin-AAPA-Bloch system with a reference, such as Walker, which discloses features of Windows security.

Walker is directed to managing a host-based firewall and describes the Windows Filtering Platform within the Windows operating system. EX1008, ¶¶[0001], [0003]. For example, Walker discloses that the Windows Filtering Platform is a firewall built into the Windows Operating System. EX1008, ¶[0070]. The firewall is used to “control a flow of information” to and from a device. EX1008, ¶[0066]. The firewall controls information by extracting data from a packet, and comparing the extracted data to a set of rules. EX1008, ¶[0142]. Rules may be based on data from the network layer, transport layer, session layer, presentation layer, or application layer. EX1008, ¶[0142]. For example, a transport layer rule may block packets including the IP address of a specific website. EX1008, ¶[0143].

A POSA would have been motivated, with a reasonable expectation of success, to use Walker’s firewall within Austin’s computer device because both Austin and Walker discuss protecting a computer using the Windows operating system. EX1003, ¶286. Austin discusses protecting a computer device using the Windows operating system. And Walker discloses a firewall built into the Windows operating system. Thus, a POSA would have reasonably expected that

Walker's firewall would work in Austin's computer device to further protect it.

EX1003, ¶¶286-288.

2. Claims 5 and 13

Claims 5 and 13 depend from claims 4 and 12, respectively, and recite “[*wherein the processor is further configured to: implement/implementing*] a host-based firewall, wherein the host-based firewall isolates the [system/first memory space and the second memory space] from communicating with one or more other devices on the network.” The Austin-Bloch-Walker combination discloses this limitation.

Walker discloses that the Windows operating system includes a built-in firewall known as the “Windows Filtering Platform” (“*a host-based firewall*”). EX1008, ¶4. A firewall regulates incoming and outgoing traffic based on a rule set (“*isolates the [system/first memory space and the second memory space] from communicating*”). EX1008, ¶3. Rules may be based on information extracted from a received packet, such as network layer transport layer, session layer, presentation layer, and/or application layer information. EX1008, ¶142. For example, an incoming packet may be allowed through the firewall based on the source IP address (“*one or more other devices on the network*”) in the packet. EX1008, ¶81.

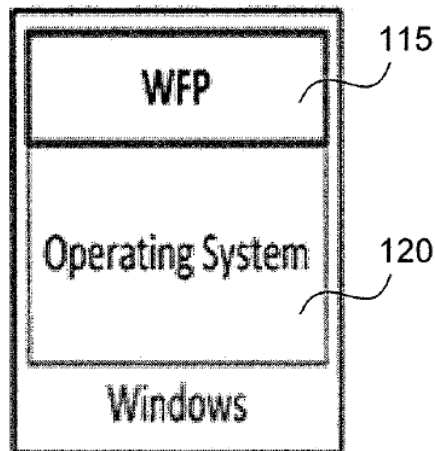


Fig. 1b

EX1008, FIG. 1b.

In the Austin-Bloch-Walker combined system, Austin’s “processor” (*“the processor”*) would implement Walker’s Windows Filtering Platform (*“host-based firewall”*) that regulates incoming and outgoing traffic (*“isolates the [system/first memory space and the second memory space] from communicating”*) based on a packet’s source IP address (*“one or more other devices on the network”*).

3. Claims 6 and 14

Claims 6 and 14 depend from claims 5 and 13, respectively, and recite *“wherein the host-based firewall blocks incoming communications from the network that are sent to the workspace.”* The Austin-Bloch-Walker combination discloses this limitation.

The '601 patent explains that the host-based firewall blocks workspace communications based on an application within the workspace, the network device sending the communication, and/or the network port used for the communication. EX1001, 6:52-60.

As noted above, Austin's primary user account ("*the workspace*") supports execution of a plurality of processes (i.e., applications). EX1005, ¶70. Additionally, Walker discloses a firewall ("*the host-based firewall*") regulating incoming traffic ("*blocking incoming communication from the network*") based on a rule set. EX1008, ¶[0003]. The rules may be based on information extracted from a network packet. EX1008, ¶[0142]. For example, a rule may be based on the application associated with the packet ("*an application in the workspace*"), a source IP address of the packet ("*the network device*"), and/or a network port ("*the port*") specified in the packet. EX1008, ¶[0142-0144].

In the Austin-Bloch-Walker combined system, Austin's computer device would implement Walker's Windows Filtering Platform ("*host-based firewall*") that regulates incoming and outgoing traffic ("*block incoming communication from the network*") based on an application executing in the workspace, the device sending the communication, or the port used for the communication ("*sent to the workspace*").

F. Ground 6: Claims 8 and 16 are obvious over Austin, AAPA, Bloch, and Franco

1. A POSA would have been motivated to combine Austin, AAPA, and Bloch with Franco based on Austin’s disclosure.

As noted above, Austin discloses a “primary clipboard . . . accessible from a primary user account,” and “a secondary clipboard . . . associated with the secondary user account.” EX1005, ¶20. In some instances, when the user initiates requests to access the clipboard, Austin discloses that, the clipboard may provide feedback to the user. For example when the computer is first started and the user attempts to paste data, the user may receive an indication when there is no data available. EX1005, ¶134. Austin also recognizes that data may be transferred between isolated and non-isolated applications via the primary and secondary clipboards. For example, data may be transferred from a non-isolated application to an isolated application, but not from an isolated application to a non-isolated application. EX1005, ¶134. Austin references “additional rules” governing when content may be transferred between the isolated and non-isolated processes EX1005, ¶137. To the extent Austin does not disclose implementation details about these “additional rules,” a POSA would have been motivated to combine the Austin-AAPA-Bloch system with a reference like Franco. EX1003, ¶297.

Similar to Austin, Franco is directed at preventing malware or other malicious software from compromising a host computer. EX1009, 1:22-29. To

address this issue, Franco discloses executing applications with “reduced privilege” so that they’re restricted from accessing portions of the host. EX1009, 2:31-36. When attempts to access a restricted area of the host are made, Franco discloses capturing explicit user input to authorize the access. EX1009, 2:36-56. For example, when the restricted area is accessed, a broker object prompts the user to interact with a dialog box or input credentials to allow the access attempt. EX1009, 5:4-11.

A POSA would have been motivated, with a reasonable expectation of success, to use Franco’s user-prompt mechanism within Austin’s client because it would amount to nothing more than combining familiar elements according to their known methods of use. *KSR*, 550 U.S. at 416. Both Austin and Franco disclose isolating processes within a computing device. Austin discloses a security module to restrict isolated applications, and “additional rules” to facilitate communication between isolated and non-isolated applications. EX1005, ¶137. Austin also discusses providing feedback to the user (i.e., indicating there is no data available to paste). Franco discloses, for example, prompting a user to facilitate communication between an application and a restricted area of the computing device. In the combined system, Austin’s security module would require Franco’s user input when data is transferred between isolated and non-isolated processes. EX1003, ¶299.

2. Claims 8 and 16

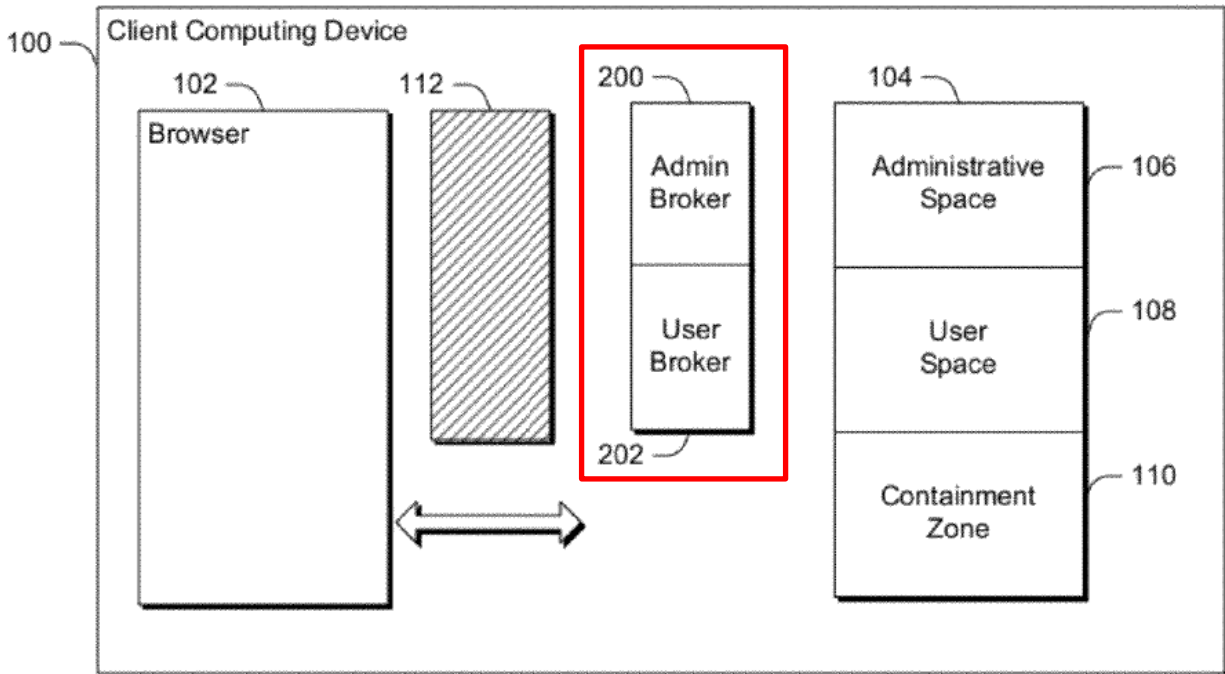
Claims 8 and 16 depend from claims 1 and 9, respectively, and recite “*wherein the internal isolation firewall prompts a user of the [system/workspace] to allow communication between the first memory space and the second memory space.*” EX1003, ¶¶300-304. The Austin-Bloch-Franco combination discloses this limitation.

As noted above, Austin’s primary user account has its own “namespace” (“*a first memory space*”). EX1005, ¶95. The primary user account has access to a “primary clipboard. EX1005, ¶20. Similarly, the namespace of the “temporary user account” (“*a second memory space*”) has its own “secondary clipboard.” EX1005, ¶20. Each clipboard is further supported by its own separate physical memory space. EX1005, ¶119. Austin’s “security module” (“*the internal isolation firewall*”) isolates tasks associated with the secondary user account from the primary user account. EX1005, ¶10.

The user (“*a user*”) initiates requests to access the clipboard. EX1005, ¶135. Either the primary or secondary clipboard is used to transfer data between the isolated and non-isolated applications (“*communication between the first memory space and the second memory space*”). EX1005, ¶¶140-141.

Franco discloses preventing applications from accessing restricted areas of a computer and requiring explicit user input when access attempts are made.

EX1009, 2:36-56. For example, when the restricted area is accessed, a broker object prompts the user with a dialog box or requiring the user to input their credentials (“*prompts a user of the [system/workspace]*”) to allow the access attempt (“*to allow communication*”). EX1009, 5:4-11.



EX1009, FIG. 2 (annotated).

In the Austin-Bloch-Franco system, Austin’s “security module” (“*internal isolation firewall*”) would display Franco’s dialog box (“*prompt a user of the [system/workspace]*”) when content is transferred between the isolated and non-isolated applications (“*to allow communication between the first memory space and the second memory space*”).

VII. NO OBJECTIVE INDICIA OF PATENTABILITY

PAN is unaware of any publicly available information about objective indicia of non-obviousness. If Croga raises arguments alleging objective indicia, PAN requests an opportunity to respond.

VIII. THE BOARD SHOULD NOT DISCRETIONARILY DENY THE PETITION.

A. *Advanced Bionics* strongly favors institution.

Applying the *Advanced Bionics* two-part framework demonstrates that denial under §325(d) is not appropriate. *Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 at 8-9 (PTAB Feb. 13, 2020) (precedential). Under part one, the same or substantially the same art or arguments were *not* previously presented to the Office, ending the inquiry in favor of institution. None of the prior-art references relied on here were considered by the Office with respect to the '601 patent. Thus, the “new, noncumulative prior art asserted in the Petition” strongly favors institution. *Oticon Medical AB v. Cochlear Ltd.*, IPR2019-00975, Paper 15 at 20 (PTAB Oct. 16, 2019) (§§II.B and II.C precedential).

B. *General Plastic* strongly favors institution.

On August 13, 2024, Cisco filed an IPR petition against the '601 patent. As explained below, the seven *General Plastic* factors heavily favor institution. *See*

General Plastic Indus. Co. v. Canon Kabushiki Kaisha, IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017) (precedential as to § II.B.4.i).

Factors 1-3 favor institution. These factors “only weigh in favor of discretionary denial when there are ‘road-mapping’ concerns under factor 3 or other concerns under factor 2.” *Code200 et al. v. Bright Data Ltd.*, IPR2022-00861 & IPR2022-00862, Paper 18 at 5 (PTAB Aug. 23, 2022) (precedential Director Review) (“*Code200*”). No such concerns exist here because this Petition relies on different prior art than in Cisco’s petition and because this Petition is being filed before the patent owner preliminary response to Cisco’s petition.

Factors 4 and 5 favor institution. This Petition has been filed within 3 weeks of Cisco’s IPR petition, even though PAN was sued more than 7 weeks after Cisco.

Factors 6 and 7 favor institution. Factor 6 is “outweighed by the Board’s mission to improve patent quality.” *Code200*, Paper 18 at 6. And factor 6 has “limited relevance.” *Id.*

C. *Fintiv* strongly favors institution.

On March 22, 2024, Patent Owner filed a complaint alleging infringement of the ’601 patent against Petitioner in the Eastern District of Texas. The *Fintiv* factors heavily favor institution. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential).

Factor 1 is neutral or favors institution. No motion for stay has been filed and the E.D. Texas court has not indicated whether it would grant any such motion.

Factor 2 favors institution. The E.D. Texas case is in early stages, and the median time to trial for civil actions in E.D. Texas is almost two years (21.6). Although jury selection for trial is currently scheduled for October 20, 2025, EX1035, the Director’s June 21, 2022 guidance recognizes, “scheduled trial dates are unreliable and often change.” *Fintiv* Guidance Memo, 8.

Factor 3 favors institution. By the time of institution, much work will remain in the parallel litigation. For example, discovery and substantive motion practice will still be ongoing. EX1035. Currently, the court has not made any substantive rulings (e.g., a claim construction order). Petitioner’s diligence filing the petition six months after filing of the complaint, within two months after the supplemental infringement contentions, within six weeks after the invalidity contentions, and six months before the statutory deadline favors institution.

Factor 4 favors institution. Petitioner stipulates that, if the Board institutes this petition, Petitioner will not raise at trial in the parallel litigation any invalidity challenges based on prior art asserted here. Additionally, this Petition challenges all claims of the ’601 patent, whereas only claims 1, 2, 4, 5, 6, 7, 9, 10, 12, 13, 14, and 15 are currently asserted in the parallel litigation. Petitioner’s stipulation and challenge to additional claims significantly mitigates the concerns under factor 4.

See Sand Revolution II, LLC v. Cont'l Intermodal Group, IPR2019-01393, Paper 24 at 12 (PTAB June 16, 2020) (informative).

Factor 6 is ***dispositive*** in favor of institution because the grounds applied here present compelling, meritorious invalidity challenges. The petition is also supported by the testimony of Dr. Bhattacharjee, adding further weight to the petition's compelling invalidity challenges.

IX. CONCLUSION

Based on the grounds specified above, *inter partes* review and cancellation of the challenged claims is respectfully requested.

Respectfully submitted,

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.

/Jonathan Tuminaro/

Jonathan Tuminaro (Reg. No. 61,327)
Ethan Goldschen (Reg. No. 80,057)
Daniel S. Block (Reg. No. 68,395)
Counsel for Petitioner Palo Alto Networks, Inc.

Date: September 16, 2024

1101 K Street, N.W., 10th Floor
Washington, D.C. 20005
(202) 371-2600

**CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME
LIMITATION, TYPEFACE REQUIREMENTS, AND TYPE STYLE
REQUIREMENTS**

1. This Petition complies with the type-volume limitation of 14,000 words, comprising 13,993 words, excluding the parts exempted by 37 C.F.R. § 42.24(a).

2. This Petition complies with the general format requirements of 37 C.F.R. § 42.6(a) and has been prepared using Microsoft® Word 2010 in 14-point Times New Roman.

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.

/Jonathan Tuminaro/

Jonathan Tuminaro (Reg. No. 61,327)
Counsel for Petitioner Palo Alto Networks, Inc.

Date: September 16, 2024

1101 K Street, N.W., 10th Floor
Washington, D.C. 20005
(202) 371-2600

CERTIFICATION OF SERVICE (37 C.F.R. §§ 42.6(e), 42.105(a))

The undersigned hereby certifies that the above-captioned **PETITION FOR
INTER PARTES REVIEW OF U.S. PATENT NO. 11,223,601** and all
supporting exhibits were served via FedEx® Express on September 16, 2024, in
their entirety on the following:

ATLANTIC IP C/O LOMBARD GELIEBTER LLP
1325 Avenue of the Americas, 28th Floor
New York, NY 10019
Correspondence Address of Record for U.S. Patent No. 11,223,601

BC LAW GROUP, P.C. C/O BRETT E. COOPER
200 Madison Avenue
New York, NY 10016
bcooper@b-clg.com
Address Known to Likely Effect Service

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.

/Jonathan Tuminaro/

Jonathan Tuminaro (Reg. No. 61,327)
Counsel for Petitioner Palo Alto Networks, Inc.

Date: September 16, 2024

1101 K Street, N.W., 10th Floor
Washington, D.C. 20005
(202) 371-2600