

**UNITED STATES PATENT AND TRADEMARK OFFICE**

---

**BEFORE THE PATENT TRIAL AND APPEAL BOARD**

---

CISCO SYSTEMS, INC.,  
Petitioner

---

IPR2024-01283  
U.S. Patent No. 11,223,601

**PETITION FOR *INTER PARTES* REVIEW  
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

TABLE OF CONTENTS

|                                                                      |    |
|----------------------------------------------------------------------|----|
| PETITIONER’S EXHIBIT LIST .....                                      | 6  |
| I. INTRODUCTION .....                                                | 9  |
| II. GROUNDS FOR STANDING.....                                        | 10 |
| III. NOTE.....                                                       | 11 |
| IV. TECHNOLOGY BACKGROUND.....                                       | 11 |
| A. Isolated Computing Environments.....                              | 11 |
| V. SUMMARY OF THE ’601 PATENT .....                                  | 12 |
| A. Overview .....                                                    | 12 |
| B. Prosecution History .....                                         | 13 |
| VI. LEVEL OF ORDINARY SKILL IN THE ART.....                          | 14 |
| VII. CLAIM CONSTRUCTION .....                                        | 15 |
| VIII. RELIEF REQUESTED AND REASONS FOR THE REQUESTED<br>RELIEF ..... | 15 |
| IX. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE.....                 | 15 |
| A. Denial under <i>Fintiv</i> is not appropriate. ....               | 15 |
| 1. No evidence regarding a stay .....                                | 15 |
| 2. Parallel proceeding trial date .....                              | 16 |
| 3. Investment in the parallel proceeding.....                        | 17 |
| 4. Overlapping issues with the parallel proceeding .....             | 17 |
| 5. Identity of parties .....                                         | 18 |

|     |                                                                                                    |    |
|-----|----------------------------------------------------------------------------------------------------|----|
| 6.  | Other circumstances .....                                                                          | 18 |
| B.  | Discretionary denial under 35 U.S.C. § 325(d) is not appropriate. ...                              | 19 |
| C.  | Discretionary denial under <i>General Plastic</i> is not appropriate.....                          | 19 |
| X.  | IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE....                                              | 20 |
| A.  | Challenged Claims and Statutory Grounds .....                                                      | 20 |
| B.  | Ground 1: Claims 1-4, 7, 9-12, and 15 are obvious in view of<br>Jeffries, LaBine, and Ishaya. .... | 21 |
| 1.  | Summary of Jeffries .....                                                                          | 21 |
| 2.  | Summary of LaBine .....                                                                            | 24 |
| 3.  | Summary of Ishaya .....                                                                            | 26 |
| 4.  | Reasons to Combine Jeffries and LaBine .....                                                       | 28 |
| 5.  | Reasons to Combine Jeffries and LaBine with Ishaya .....                                           | 32 |
| 6.  | Claim 1 .....                                                                                      | 36 |
| 7.  | Claim 2 .....                                                                                      | 74 |
| 8.  | Claim 3 .....                                                                                      | 81 |
| 9.  | Claim 4 .....                                                                                      | 82 |
| 10. | Claim 7 .....                                                                                      | 83 |
| 11. | Claim 9 .....                                                                                      | 85 |
| 12. | Claim 10 .....                                                                                     | 88 |
| 13. | Claim 11 .....                                                                                     | 89 |
| 14. | Claim 12 .....                                                                                     | 90 |

|      |                                                                                                          |     |
|------|----------------------------------------------------------------------------------------------------------|-----|
| 15.  | Claim 15 .....                                                                                           | 90  |
| C.   | Ground 2: Claims 5-6 and 13-14 are obvious in view of Jeffries,<br>LaBine, Ishaya, and Cucinotta. ....   | 91  |
| 1.   | Summary of Cucinotta. ....                                                                               | 91  |
| 2.   | Reasons to Combine Cucinotta with Jeffries, LaBine,<br>and Ishaya. ....                                  | 93  |
| 3.   | Claim 5 .....                                                                                            | 95  |
| 4.   | Claim 6 .....                                                                                            | 99  |
| 5.   | Claim 13 .....                                                                                           | 100 |
| 6.   | Claim 14 .....                                                                                           | 102 |
| D.   | Ground 3: The combination of Jeffries, LaBine, Ishaya, and Levy<br>renders obvious claims 8 and 16. .... | 102 |
| 1.   | Summary of Levy .....                                                                                    | 102 |
| 2.   | Reasons to Combine Levy with Jeffries, LaBine, and<br>Ishaya. ....                                       | 103 |
| 3.   | Claim 8 .....                                                                                            | 106 |
| 4.   | Claim 16 .....                                                                                           | 107 |
| XI.  | CONCLUSION .....                                                                                         | 108 |
| XII. | MANDATORY NOTICES .....                                                                                  | 109 |
| A.   | Real Party-in-Interest .....                                                                             | 109 |
| B.   | Related Matters .....                                                                                    | 109 |
| C.   | Lead and Back-up Counsel and Service Information .....                                                   | 109 |
|      | CERTIFICATE OF WORD COUNT .....                                                                          | 111 |

|                              |     |
|------------------------------|-----|
| CERTIFICATE OF SERVICE ..... | 112 |
|------------------------------|-----|

**PETITIONER’S EXHIBIT LIST**

|         |                                                                                                                                                                                 |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ex.1001 | U.S. Patent No. 11,223,601 to Coleman et al.                                                                                                                                    |
| Ex.1002 | Prosecution History of U.S. Patent No. 11,223,601.                                                                                                                              |
| Ex.1003 | Declaration of Dr. Prashant Shenoy under 37 C.F.R. § 1.68.                                                                                                                      |
| Ex.1004 | <i>Curriculum Vitae</i> of Dr. Shenoy.                                                                                                                                          |
| Ex.1005 | U.S. Patent No. 10,885,189 to Jeffries et al. (“Jeffries”).                                                                                                                     |
| Ex.1006 | U.S. Patent Publication No. 2009/0292999 to LaBine et al. (“LaBine”).                                                                                                           |
| Ex.1007 | U.S. Patent Publication No. 2014/0282889 to Ishaya et al. (“Ishaya”).                                                                                                           |
| Ex.1008 | U.S. Patent Publication No. 2016/0048406 to Cucinotta et al. (“Cucinotta”).                                                                                                     |
| Ex.1009 | U.S. Patent Publication No. 2007/0136579 to Levy et al. (“Levy”).                                                                                                               |
| Ex.1010 | U.S. Patent No. 6,167,432 to Jiang (“Jiang”).                                                                                                                                   |
| Ex.1011 | Prevelakis, “Sandboxing Applications,” presented at USENIX Annual Technical Conference 2001 (“Prevelakis”).                                                                     |
| Ex.1012 | Goldberg, A Secure Environment for Untrusted Helper Applications (Confining the Wily Hacker), Proceedings of the Sixth USENIX UNIX Security Symposium (July 1996) (“Goldberg”). |
| Ex.1013 | U.S. Patent No. 5,974,549 to Golan (“Golan”).                                                                                                                                   |
| Ex.1014 | U.S. Patent No. 5,987,611 to Freund (“Freund”).                                                                                                                                 |
| Ex.1015 | U.S. Patent Publication No. 2002/0178375 to Whittaker et al. (“Whittaker”).                                                                                                     |
| Ex.1016 | U.S. Patent No. 6,292,874 to Barnett (“Barnett”).                                                                                                                               |

|         |                                                                                                                                             |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Ex.1017 | U.S. Patent No. 5,067,072 to Talati et al. (“Talati”)                                                                                       |
| Ex.1018 | U.S. Patent Publication No. 2006/0112384 to Frank et al. (“Frank”)                                                                          |
| Ex.1019 | U.S. Patent No. 8,326,872 to Zwilling et al. (“Zwilling”)                                                                                   |
| Ex.1020 | U.S. Patent Publication No. 2009/0259720 to Heins et al. (“Heins”)                                                                          |
| Ex.1021 | U.S. Patent No. 6,226,677 to Slemmer (“Slemmer”)                                                                                            |
| Ex.1022 | U.S. Patent Publication No. 2013/0191880 to Conlan et al. (“Conlan”)                                                                        |
| Ex.1023 | U.S. Patent Publication No. 2013/0097706 to Titonis et al. (“Titonis”)                                                                      |
| Ex.1024 | U.S. Patent Publication No. 2015/0113544 to Yu (“Yu”)                                                                                       |
| Ex.1025 | U.S. Patent Publication No. 2009/0225767 to Song et al. (“Song”)                                                                            |
| Ex.1026 | U.S. Patent Publication No. 2005/0223118 to Tucker et al. (“Tucker”)                                                                        |
| Ex.1027 | U.S. Patent Publication No. 2002/0010800 to Riley et al. (“Riley”)                                                                          |
| Ex.1028 | U.S. Statutory Invention Registration H1944 H to Cheswick et al. (“Cheswick”)                                                               |
| Ex.1029 | U.S. Patent Publication No. 2011/0078550 to Nabutovsky (“Nabutovsky”)                                                                       |
| Ex.1030 | Complaint - <i>Croga Innovations Ltd. v. Cisco Systems, Inc.</i> EDTX-2-24-cv-00065.                                                        |
| Ex.1031 | Amended Docket Control Order - <i>Croga Innovations Ltd. v. Cisco Systems, Inc.</i> , No. 2-24-cv-00065, Dkt. 42 (E.D. Tex. July 18, 2024). |
| Ex.1032 | Federal Court Trial Statistics – June 30, 2024.                                                                                             |
| Ex.1033 | List of patent trials scheduled before Judge Gilstrap (E.D. Tex.) on October 20, 2025.                                                      |

|         |                                                                                                                                                                                                                                                                                                                                                                              |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ex.1034 | Azab, SICE: A Hardware-Level Strongly Isolated Computing Environment for x86 Multi-core Platforms, (“Azab”).                                                                                                                                                                                                                                                                 |
| Ex.1035 | U.S. Patent Publication No. 2016/0088155 to Patel (“Patel”)                                                                                                                                                                                                                                                                                                                  |
| Ex.1036 | U.S. Patent Publication No. 2013/0110999 to Kopasz et al. (“Kopasz”)                                                                                                                                                                                                                                                                                                         |
| Ex.1037 | U.S. Patent Publication No. 2016/0381072 to Todd et al. (“Todd”)                                                                                                                                                                                                                                                                                                             |
| Ex.1038 | Revised Infringement Contentions – Croga Innovations Ltd. v. Cisco Systems, Inc., No. 2-24-cv-00065, Dkt. 52 (E.D. Tex. July 16, 2024).                                                                                                                                                                                                                                      |
| Ex.1039 | 2001 FREENIX Track Technical Program – Abstract for “Sandboxing Applications” (as captured by the Wayback Machine on January 26, 2004, <a href="https://web.archive.org/web/20040126143558/https://www.usenix.org/event/usenix01/freenix01/prevelakis.html">https://web.archive.org/web/20040126143558/https://www.usenix.org/event/usenix01/freenix01/prevelakis.html</a> ) |
| Ex.1040 | FREENIX Track: 2001 USENIX Annual Technical Conference (as captured by the Wayback Machine on September 15, 2003, <a href="https://web.archive.org/web/20030915151057/https://www.usenix.org/event/usenix01/freenix01/technical.html">https://web.archive.org/web/20030915151057/https://www.usenix.org/event/usenix01/freenix01/technical.html</a> )                        |
| Ex.1041 | “Using the Wayback Machine,” <a href="https://help.archive.org/help/using-the-wayback-machine/">https://help.archive.org/help/using-the-wayback-machine/</a> .                                                                                                                                                                                                               |

## I. INTRODUCTION

Pursuant to 35 U.S.C. §§ 311, 314(a), and 37 C.F.R. § 42.100, Cisco Systems, Inc. (“Petitioner”) respectfully requests that the Board review and cancel as unpatentable under 35 U.S.C. §103 claims 1-16 (the “Challenged Claims”) of U.S. Patent No. 11,223,601 (“’601 patent,” Ex.1001).

The ’601 patent claims executing a “*multi-user interactive software application*” in an authenticated “*isolated computing environment*.” Ex.1001, 22:49-59 (claim 1). Doing so “protect[s] the host computer” because malware that may be received by the software application is “prevented from access[ing] any additional resources of the host computer.” Ex.1001, 4:12-15, 7:2-3. The ’601 patent identifies Cisco’s Webex video conferencing software as an example of its “*multi-user interactive software application*,” claimed. Ex.1001, 4:45-47 (“The collaboration software...is commercially available, for example, such as **WebEx**.”). It describes an isolated computing environment as a “sandbox container process,” Ex.1001, 5:11-20, and a “virtual machine,” Ex.1001, 5:8-10.

None of these concepts, nor their combination, were new. At the time of filing, those skilled in the art had been executing applications in isolated computing environments for decades. *See, e.g.*, Ex.1012, Abstract (published 1996) (“creat[ing] a secure environment to contain untrusted...applications” and “restricting the program’s access to the operating system”); Ex.1015, [0011]

(published 2002) (“protecting a host computer from malicious mobile code” with “sandbox-based protection”); Ex.1013, Abstract (published 1999); Ex.1014, 22:15-19 (published 1999).

In short, the ’601 patent merely proposed placing known video conferencing software in one of these well-known isolated computing environments. But for years before the ’601 patent was filed, those skilled in the art understood that *any* type of application could be executed in an isolated computing environment. Ex.1005, 9:26-32, 10:54-63; Ex.1011, 8; Ex.1012, 10 (discussing “sandboxing” for application isolation in 1996). There was nothing patentable about applying a known software execution technique to running conferencing software, like “Webex,” that was “commonly known.” Ex.1035, [0044]. Doing so is the textbook definition of obviousness: combining “known methods” to yield “predictable results.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 401 (2007).

For these reasons, Petitioner respectfully requests that the Board review and cancel as unpatentable the Challenged Claims of the ’601 patent.

## **II. GROUNDS FOR STANDING**

Petitioner certifies that the ’601 Patent is eligible for IPR and that Petitioner is not barred or estopped from requesting IPR challenging the claims. 37 C.F.R. §42.104(a).

### **III. NOTE**

Petitioner cites to exhibits' original page numbers. Emphasis in quoted material has been added.

### **IV. TECHNOLOGY BACKGROUND**

#### **A. Isolated Computing Environments**

The '601 patent is primarily directed to isolated computing environments. *See* Ex.1001, Abstract, 1:58-60. An isolated computing environment may be referred to as a virtual machine, container, or sandbox. *See, e.g.*, Ex.1001, 5:41-43; Ex.1005, Abstract, 6:9-12; Ex.1007, [0041]; Ex.1008, Abstract; Ex.1011, 1; *see also* Ex.1039 (showing Ex.1011 publication information); Ex.1040, 2; Ex.1041, 5 (explaining archive.org website addresses formatted as “yyyymmddhhmmss” (year, month, day, hour, minute, second)). Isolated computing environments quarantine a region of memory from the remainder of a host computer system. Ex.1001, 3:8-17, 11:65-12:3; Ex.1005, 16:26-34, 12:7-22; Ex.1007, [0041]; Ex.1016, Abstract; Ex.1017, 6:9-12. Applications that access untrusted resources are executed in isolated computing environments. Ex.1005, 10:54-63. If an application is exposed to malware, the malware effects are contained within the isolated computing environment, preventing infestation. Ex.1005, 16:4-16, 16:37-42.

In some instances, an isolated computing environment may behave like a

virtual machine and appear on the network to be distinct from the host machine. The isolated computing environment would have its own identifying information, such as a MAC address or IP address. Ex.1007, [0043]. To connect to a local network or access the internet, including accessing trusted or untrusted sources, the isolated computing environment authenticates itself using its assigned identifier just as a physical machine would authenticate itself to a typical network. Ex.1007, [0179]-[0184]; Ex.1018, [0003]; Ex.1019, [0018]; Ex.1020, [0180].

## **V. SUMMARY OF THE '601 PATENT**

### **A. Overview**

The '601 patent describes protecting a host computer from “malware” by isolating multi-user interactive software. Ex.1001, 4:11-12, 36-42. The software allows one or more remote machines “to control (e.g., access, view, interact with, download from, upload to, and/or modify) an application running on [a] host computer system.” Ex.1001, 2:19-23. An internal isolation firewall enforces the isolation. Ex.1001, 1:67-2:6.

Communications with untrusted sources are routed through a web proxy. Ex.1001, 14:46-50 (sending traffic to “untrusted network destinations using...a web proxy 336”). The isolated computing environment is authenticated “via an authentication device.” Ex.1001, 2:8-17, 6:14-18.

**B. Prosecution History**

The application for the '601 patent was filed on September 25, 2018 and claims priority to a provisional filed on September 28, 2017. The as-filed application included 17 total claims, with claims 1, 9, and 17 being independent. Ex.1002, 1-37.

After a restriction requirement, the first Office action rejected claims 1-16 for nonstatutory double patenting and obviousness over U.S. Patent No. 8,966,464 to Christopher ("Christopher") in view of U.S. Patent Publication 2014/0040979 to Barton ("Barton"). Ex.1002, 97-112. In response, the applicant added language relating to "send[ing] data to an untrusted destination" to claims 1 and 9. Ex.1002, 131-136.

A final Office action maintained the rejections while holding the nonstatutory double patenting rejection in abeyance. In response, the applicant reworded the claim language related to authentication in claims 1 and 9. Ex.1002, 157-162, 169-171.

In continued prosecution, the examiner issued a non-final Office action rejecting claims 1-8 under 35 U.S.C. §112. The action also rejected claims 1-16 as obvious over a non-patent reference authored by Azab ("Azab") (Ex.1034) in view of U.S. Patent No. 8,566,398 to Barnier ("Barnier"). Ex.1002, 176-187. In response, the applicant added claim language reciting "a memory," "a processor,"

and “a host operating system.” Ex.1002, 196-201, 216-218.

The examiner then issued an Ex Parte Quayle action indicating that claims 1-16 were in condition for allowance pending the cancellation of claim 17.

Ex.1002, 249-254. The Ex Parte Quayle action indicated that the claim language “an isolated computing environment that uses the host operating system” was not described in the prior art. Ex.1002, 252.

The Patent Office did not address the nonstatutory double patenting rejection previously held in abeyance, and no terminal disclaimer was ever filed. The applicant cancelled claim 17, Ex.1002, 258, and received a notice of allowance. Ex.1002, 281-285.

## **VI. LEVEL OF ORDINARY SKILL IN THE ART**

A Person of Ordinary Skill in the Art (“POSITA”) in September 2017 would have been someone knowledgeable and familiar with network security and application isolation techniques as those are applied in the context of network security available in the 2010s. That person would have a bachelor’s degree in computer science, computer engineering, electrical engineering, or equivalent training, and approximately two years of experience working in the field of network security. Additional work experience can substitute for educational background, and vice versa. Ex.1003, ¶¶20-22.

## VII. CLAIM CONSTRUCTION

Claim terms in IPR are construed according to their “ordinary and customary meaning” to those of skill in the art. 37 C.F.R. §42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (en banc). Petitioner submits that, for the purposes of this proceeding and the grounds presented herein, no claim term requires express construction. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017); Ex.1003, ¶¶36-37.

## VIII. RELIEF REQUESTED AND REASONS FOR THE REQUESTED RELIEF

Petitioner asks that the Board institute a trial for *inter partes* review and cancel the Challenged Claims.

## IX. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE

### A. Denial under *Fintiv* is not appropriate.

The six factors considered for § 314 discretionary denial strongly favor institution given Petitioner’s prompt filing and the compelling merits of this case. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential).

#### 1. No evidence regarding a stay

No motion to stay has been filed, so the Board should not infer the outcome of such a motion. *Sand Revolution II LLC v. Continental Intermodal Group – Trucking LLC*, IPR2019-01393, Paper 24 at 7 (PTAB June 16, 2020)

(informative); *see also Dish Network L.L.C. v. Broadband iTV, Inc.*, IPR2020-01359, Paper 15 (Feb. 12, 2021) (“It would be improper to speculate, at this stage, what the Texas court might do regarding a motion to stay...”). Thus, this factor is neutral on discretionary denial.

## **2. Parallel proceeding trial date**

The co-pending district court case was filed in the Eastern District of Texas on February 1, 2024. The current statistics for the Eastern District of Texas show a median time-to-trial of 21.6 months. Ex.1032, 35. The estimated trial date is late December 2025.<sup>1</sup> The Board’s final written decision is expected afterwards, in early February 2026. Given the proximity of the Board’s expected Final Written Decision and the estimated trial date, this factor is at least neutral. *See e.g., Cisco Systems Inc., v. Video Solutions PTE. Ltd.*, IPR2024-00194, Paper 7 (PTAB March

---

<sup>1</sup> The docket control order sets the case for trial on October 20, 2025. Ex.1031, 1. But “a fast-paced schedule is not particularly relevant...where, like here, the forum itself has not historically resolved cases so quickly.” *In re Apple Inc.*, 979 F.3d 1332, 1344 (Fed. Cir. 2020). Notably, the court already has five more patent trials—plus four other trials—scheduled to begin October 20, 2025. *See* Ex.1033.

28, 2024) (granting institution and finding factor two neutral where “trial is likely to begin within approximately one month of [the Board’s] final written decision”).

### **3. Investment in the parallel proceeding**

The co-pending litigation is in its early stages, with major activities including a *Markman* hearing and the close of discovery scheduled for after the Board’s expected decision on institution. *See* Ex.1031, 4 (*Markman* hearing on April 30, 2025; close of fact discovery on May 23, 2025; close of expert discovery on July 8, 2025). Moreover, Petitioner has acted with diligence to prepare this Petition approximately six months after Petitioner was served with a complaint, and within weeks of receiving revised infringement contentions (identifying the asserted claims). Ex.1038, 2. Under *Fintiv*, Petitioner’s prompt filing “weigh[s] against exercising the authority to deny institution.” *Fintiv*, Paper 11 at 11 (“If the evidence shows that the petitioner filed the petition expeditiously, such as promptly after becoming aware of the claims being asserted, this fact has weighed against exercising the authority to deny institution under NHK.”).

### **4. Overlapping issues with the parallel proceeding**

Petitioner stipulates that if trial is instituted, it will not pursue the same grounds in the district court litigation. *See Sand Revolution II, LLC v. Cont’l Intermodal Group-Trucking LLC*, IPR2019-01393, Paper 24 at 12 (PTAB June 16, 2020). This factor favors institution.

**5. Identity of parties**

Petitioner is a defendant in the litigation, which is common. Accordingly, this factor should not be a basis for denying institution.

**6. Other circumstances**

The prior art presented in this Petition presents a compelling case on the merits. The claims of the '601 patent were allowed after the Applicant argued that the isolated container in the prior art (Azab, Ex.1034) “does not rely on any software component in the host environment. (i.e., an OS or a hypervisor).” Ex.1002, 217. The isolated computing environment of the amended claims “uses the host operating system” Ex.1002, 217 (quoting claim 1).

Jeffries describes an isolated “process container” that uses the host operating system “using namespace isolation.” Ex.1005, 10:22-34; *see* Ex.1001, 5:57-6:8 (similarly describing “namespace isolation”). Jeffries’ isolated containers access “shared parts of host operating system 102.” Ex.1005, 10:28-34; Ex.1003, ¶¶129-131. As explained in more detail below, the remaining limitations were also known and would have been obvious to use in combination.

The Board’s policy is to “not deny institution of an IPR or PGR under *Fintiv* (i) when a petition presents compelling evidence of unpatentability.” Director Vidal Memorandum (June 21, 2022) (“Memo”), 9. “Compelling, meritorious challenges are those in which the evidence, if unrebutted in trial, would plainly

lead to a conclusion that one or more claims are unpatentable by a preponderance of the evidence.” Memo, 4. Here, the Petition shows that the concepts claimed were taught by Jeffries in a predictable combination with LaBine and Ishaya. The prosecution examiner allowed the claims without evaluating their unpatentability in view of any of these prior art references.

Thus, the evidence of unpatentability is compelling here, and the PTAB should not deny institution under *Fintiv*.

**B. Discretionary denial under 35 U.S.C. § 325(d) is not appropriate.**

The challenges presented in this petition are neither cumulative nor redundant to the prosecution of the ’601 patent. The Examiner did not discuss or apply any of the references relied upon in this petition. The challenges in this petition are non-cumulative because the prior art presented describes the limitations that were distinguished by the Applicant to gain allowance. Accordingly, the Board should not exercise its discretion to deny this petition under § 325(d).

**C. Discretionary denial under *General Plastic* is not appropriate.**

The ’601 patent has not been challenged in any prior IPR petition, so none of the *General Plastic* discretionary institution factors apply. *See General Plastic Indus. Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at 16 (PTAB Sept. 6, 2016) (Section II.B.4.i. precedential).

## **X. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE**

### **A. Challenged Claims and Statutory Grounds**

| <b>Grounds</b> | <b>Claims</b>        | <b>Basis</b>                                                 |
|----------------|----------------------|--------------------------------------------------------------|
| #1             | 1-4, 7, 9-12, and 15 | 35 U.S.C. § 103 over Jeffries, LaBine, and Ishaya            |
| #2             | 5-6 and 13-14        | 35 U.S.C. § 103 over Jeffries, LaBine, Ishaya, and Cucinotta |
| #3             | 8 and 16             | 35 U.S.C. § 103 over Jeffries, LaBine, Ishaya, and Levy      |

U.S. Patent No. 10,885,189 to Jeffries et al. (“Jeffries”) was filed on May 22, 2017 and is prior art under 35 U.S.C. §102(a)(2).

U.S. Patent Publication No. 2009/0292999 to LaBine (“LaBine”) was published on November 26, 2009 and is prior art under 35 U.S.C. §102(a)(1).

U.S. Patent Publication No. 2014/0282889 to Ishaya et al. (“Ishaya”) was published on September 18, 2014 and is prior art under 35 U.S.C. §102(a)(1).

U.S. Patent Publication No. 2016/0048406 to Cucinotta et al. (“Cucinotta”) was published on February 18, 2016 and is prior art under 35 U.S.C. §102(a)(1).

U.S. Patent Publication No. 2007/0136579 to Levy et al. (“Levy”) was published on June 14, 2007 and is prior art under 35 U.S.C. §102(a)(1).

**B. Ground 1: Claims 1-4, 7, 9-12, and 15 are obvious in view of Jeffries, LaBine, and Ishaya.**

**1. Summary of Jeffries**

Like the '601 patent, Jeffries protects a host computer by isolating applications that access untrusted sources. *See* Ex.1005, Abstract. Jeffries' isolated containers 122 are illustrated in Figure 1 below. Ex.1003, ¶41.

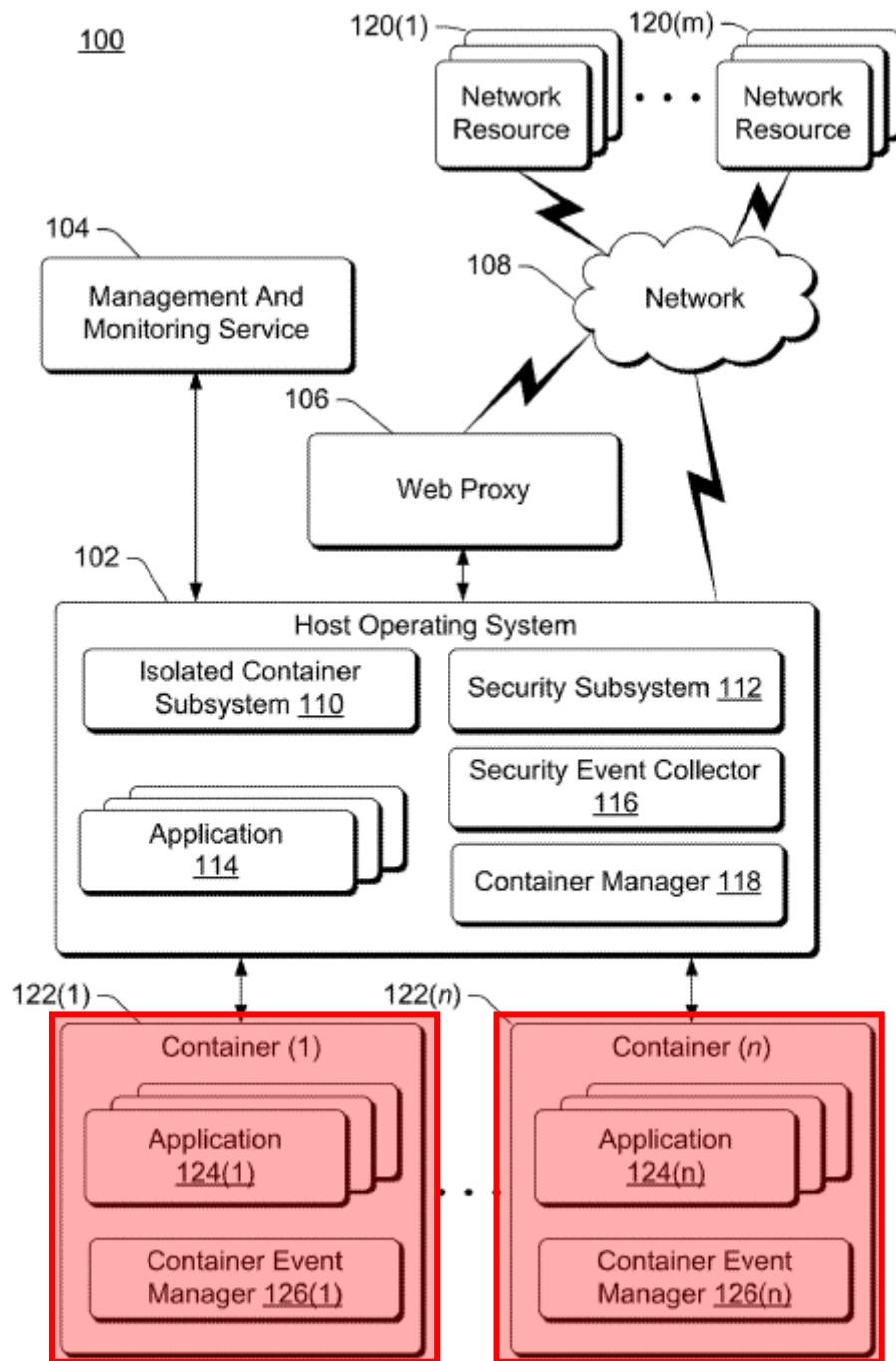


**Fig. 1**

**Ex.1005, Fig. 1 (annotated)**

Jeffries' "applications 124" are "isolat[ed]" to "protect[]" the computing

device from attacks.” Ex.1005, 10:54-63, 2:45-49; Ex.1003, ¶¶42-43.

Jeffries’ system includes a “host operating system 102” that is isolated from the containers 122. Ex.1005, 6:44-46. “The host operating system 102” includes “one or more applications 114.” Ex.1005, 8:55-58. The “[a]pplications 114 may include a web browser application...such as e-mail applications, word processing applications, spreadsheet applications, visual presentation applications, and the like.” Ex.1005, 9:28-33; Ex.1003, ¶44.

Jeffries’ containers 122 include applications 124 and are isolated from the host operating system by a container manager 118. The container manager 118 “quarantine[s] the container 122 and prevent[s] the container 122 from further accessing resources.” Ex.1005, 16:26-31. This “prevents malicious activity occurring in the container from gaining file system access.” Ex.1005, 16:31-37; *see also* 12:7-22; Ex.1003, ¶45. Applications 124 are “useable to access untrusted resources” because they are isolated from the remaining system. Ex.1005, 10:54-63; *see also* 3:48-51; Ex.1003, ¶46.

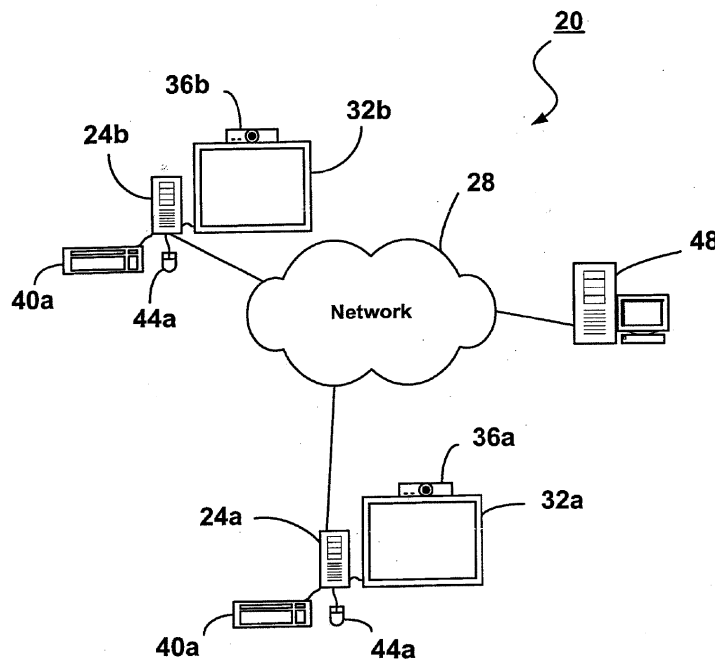
Jeffries’ system also includes a web proxy 106 between the host computer and external network resources 120. *See* Ex.1005, Fig. 1, 6:44-57; Ex.1003, ¶47.

Jeffries is directed to the same technical field as the ’601 patent. The ’601 patent relates to “isolat[ing] applications” that “have access to...untrusted Internet destinations.” Ex.1001, 2:10-11. Jeffries similarly pertains to “an application

running in a[n isolated] container” that accesses “untrusted resources.” Ex.1005, Abstract, 2:42-49; Ex.1003, ¶48. In addition, Jeffries addresses the same problem as the ’601 patent: improving security of a computer against potential security threats from untrusted sources. *See* Ex.1001, 2:8-11; Ex.1005, 1:6-9; Ex.1003, ¶49.

## 2. Summary of LaBine

LaBine describes video conferencing software called the “SMART Bridgit Conferencing Software.” Ex.1006, [0045], [0043], Fig. 1. Figure 1 shows a “computing environment 20” including “computers 24a, 24b communicating over a communication network 28.” Ex.1006, [0043]; Ex.1003, ¶50.



Ex.1006, Fig. 1.

In a video conference, “one personal computer” is “a host, or ‘presenter’

computer and the remaining personal computers [are] client, or ‘participant’ computers.” Ex.1006, [0045]; Ex.1003, ¶51. A client computer requests remote control of the host computer with a “REMOTE CONTROL request.” Ex.1006, [0078]. If approved, the client computer may “input events on the client computer..., such as mouse, keyboard and pen events” and the host computer “treats the remote user input events as though they were input locally.” Ex.1006, [0084]; Ex.1003, ¶52.

LaBine is also directed to the same technical field as the ’601 patent. The ’601 patent describes *a multi-user interactive software application*, Ex.1001, 22:55 (Claim 1), that allows *a remote machine to control an application*, Ex.1001, 22:65-66 (Claim 2). LaBine grants a client computer “remote control of the host computer” during a web conference. Ex.1006, [0078]-[0084]. LaBine addresses security by protecting transmitted data through “SSL connections,” Ex.1006, [0046], and positioning a “frame 120 to exclude sensitive controls from the shared region, thereby ensuring that other participants are not granted access to them,” Ex.1006, [0048]; Ex.1003, ¶53.

LaBine addresses the same problem of the ’601 patent: improving collaborative software applications by allowing users of client computers to remotely control host computers while providing security mechanisms. *See* Ex.1001, 2:19-23 (“allow[ing] one or more remote machines to control...the host

computer system”); Ex.1006, Abstract (“enabling remote control of a host computer”); Ex.1003, ¶54.

### 3. Summary of Ishaya

Ishaya describes authenticating “virtual machine instances.” Ex.1007, Abstract. Ishaya’s “system 210...host[s] one or more virtual machines.” Ex.1007, [0035]; Ex.1003, ¶55. Ishaya’s techniques apply to “virtual machines” and “logical container[s].” Ex.1007, [0041]; Ex.1003, ¶¶55-56.

In Ishaya, each virtual machine is assigned “identification data,” such as a “(MAC) address,” or “(IP) address.” Ex.1007, [0043], [0053]. Ishaya describes creating and authenticating a virtual machine with Figure 7, below. *See* Ex.1007, [0177]- [0187]; Ex.1003, ¶¶57-58.

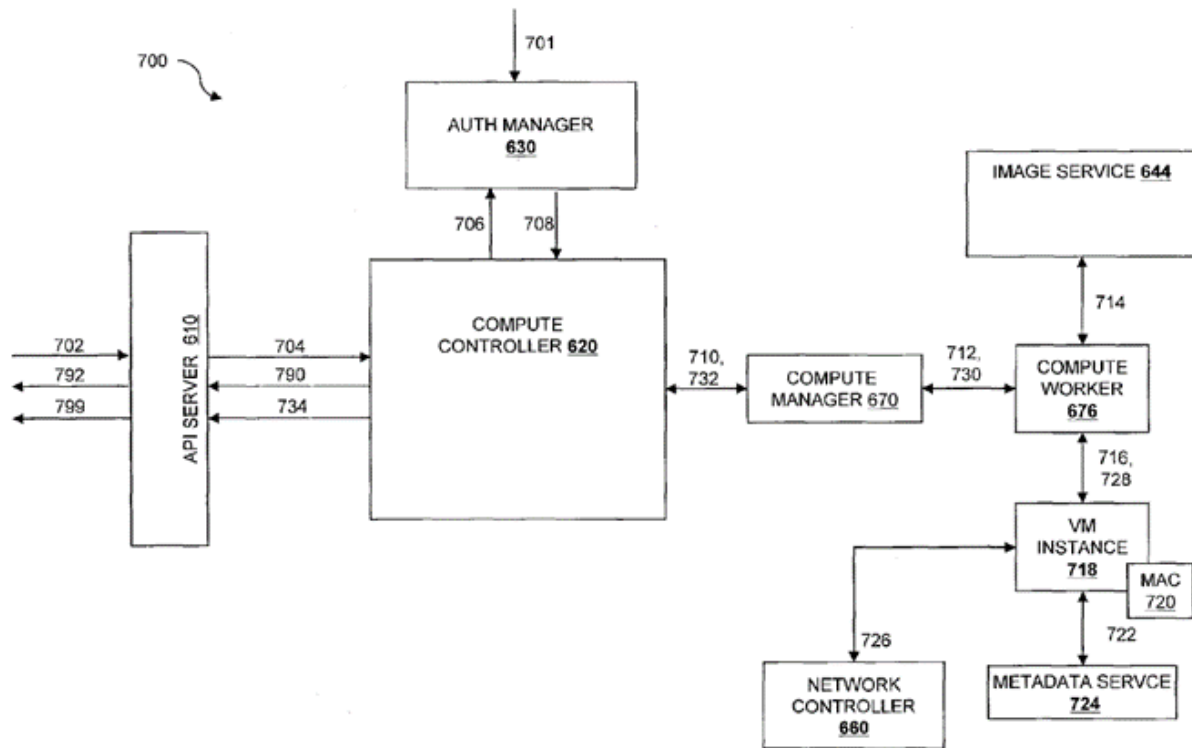


FIGURE 7

**Ex.1007, Fig. 7**

The virtual machine in Ishaya is assigned “a MAC address 720” and/or “an IP address.” Ex.1007, [0181], [0185]. The virtual machine then uses this identity information to authenticate itself to “the Metadata Service 724” and “the Network Controller 660.” Ex.1007, [0184]-[0185], [0192] (“At step 820, the VM uses the MAC address to authenticate itself.”), Fig. 8; Ex.1003, ¶59.

Ishaya is analogous art to the '601 patent. The '601 patent describes an “authentication device” that only allows an isolated computing environment to “communicate with the untrusted network destinations...after [it] has [been]

authenticated.” Ex.1001, 2:13-17, 6:14-21. Ishaya similarly describes authenticating virtual machines “using identity information.” Ex.1007, [0001]. Like the ’601 patent, virtual machines and containers in Ishaya may be isolated. Ex.1007, [0041], [0194], Fig. 2. Thus, Ishaya is in the same field of endeavor and is reasonably pertinent to a particular problem with which the inventor of the ’601 patent was involved (i.e., container authentication). Ex.1003, ¶60.

#### **4. Reasons to Combine Jeffries and LaBine**

A POSITA would have combined Jeffries with LaBine because both improve the security of a host computer when accessing untrusted resources. *See* Ex.1005, 2:46-49, 10:54-63 (“Applications 124” within isolated containers “are thus useable to access untrusted resources.”); Ex.1006, [0004], [0050] (describing “[w]eb conferencing software” in which “viewers are given remote control access”); Ex.1003, ¶61.

A POSITA would have been motivated to combine Jeffries and LaBine to execute a web conferencing application, like LaBine’s, within an isolated container, as described by Jeffries. Doing so would isolate the host computer from any malicious code potentially introduced through the web conferencing application. *See* Ex.1005, 2:42-49, 10:54-63; Ex.1003, ¶¶62-63.

Jeffries’ isolated applications 124 can be of any type, including any “applications that are executable by one or more processors.” Ex.1005, 9:26-27.

Examples include “web browser application[s],” “email applications, spreadsheet applications, visual presentation applications, and the like.” Ex.1005, 9:28-33 (listing types of applications 114); *see* 10:54-58 (“applications 124(1),..., 124(n) correspond to...applications 114”)); Ex.1003, ¶64.

A POSITA would have understood that Jeffries’ application-isolation techniques were applicable to **any** type of application, including a web conferencing application such as LaBine’s. *See* Ex.1005, 9:26-33, 10:56-58; Ex.1003, ¶65. Jeffries’ application-isolation techniques would have been obvious to apply to “well-known” software, such as a remote conferencing or control application. Ex.1036, [0004]; Ex.1003, ¶65.

LaBine’s video conferencing software allows participants to control the host user’s computer. *See* §X.B.2; Ex.1006, Abstract, [0078]-[0084]. This collaborative software may transmit or receive “sensitive information,” Ex.1006, [0046], as well as expose sensitive and/or personal information of the host computer. Ex.1006, [0050]; Ex.1003, ¶66. Thus, LaBine’s collaborative software accesses an untrusted source by allowing a remote user, including unknown users, to control the host computer. Ex.1006, [0084]. Remote users may control the host computer to download and execute malicious content, access sensitive or confidential data, or perform other nefarious acts. Accordingly, a POSITA would have recognized that a user applying LaBine’s web conferencing application would readily benefit from

Jeffries' methods of isolating applications in containers. Ex.1003, ¶67.

LaBine simply describes a specific type of application that may be executed in Jeffries' containers 122. *See* Ex.1005, 9:26-33; Ex.1006, Abstract, [0078]-[0084]. Jeffries' applications 124 "access untrusted resources." Ex.1005, 10:59-63. LaBine's web conferencing application accesses untrusted resources. Ex.1006, [0084]. Thus, LaBine's web conferencing is the type of application Jeffries anticipates as being executed within its isolated containers 122. Ex.1003, ¶68.

Combining LaBine with Jeffries would have been a simple combination of prior art elements according to known methods to yield predictable results. In the combination, LaBine's web conferencing application would be isolated within a container as described in Jeffries. *See* Ex.1006, [0045], [0084]; Ex.1005, Abstract, 2:42-49. This combination would have been predictable with an expectation of success because Jeffries describes isolating any type of application. *See* Ex.1005, 9:28-33, 10:54-58; *see also* 3:48-51; Ex.1003, ¶69.

In addition, multiple advantages would motivate a POSITA to combine Jeffries with LaBine. The combination advantageously improves the security of a computer system that allows remote control by participating computers. A POSITA would have recognized that a web conferencing application, such as LaBine's, would allow users to collaborate with team members within an organization. With the added layer of security provided by Jeffries' isolated

containers, users could use these same collaborative techniques with unknown participants including sharing files, presenting media content, or granting remote access to users outside of their organization, while protecting the host computer. Ex.1003, ¶70.

Further, isolating LaBine’s interactive remote-control software in a container 122 would prevent it “from accessing any resources or data outside the container.” *See* Ex.1005, 5:51-54, 12:17-22. Thus, the effect of executed code would be contained within the isolated container and would not affect other parts of the host computer. *See* Ex.1005, 16:26-31. If the isolated container were negatively impacted, for example by a malware infection, a “corrective action” would include “terminating the container 122 and rebuilding a new container”—without impacting the host computer. Ex.1005, 16:37-42; Ex.1003, ¶71.

A POSITA would have had a reasonable expectation of success in making the combination. While Jeffries’ techniques apply to any kind of application, *see* Ex.1005, 9:26-33, Jeffries specifically contemplates isolating applications that access untrusted resources, *see* Ex.1005, 10:54-63, 3:48-51. LaBine’s web conferencing application is this type of application because it allows remote access by untrusted users. Ex.1006, [0084]. In addition, Jeffries’ container isolation policies are flexible enough to ensure that any application has access to the necessary resources for execution. *See* Ex.1005, 15:26-30 (describing “allocat[ing]

resources that may be required by container subsystems 206”). Running LaBine’s web conferencing application in Jeffries’ isolated containers would have been within the skillset of a POSITA. Ex.1003, ¶72.

For all these reasons, a POSITA would be motivated to implement a multi-user interactive software application, like LaBine’s software, in an isolated container, as described by Jeffries. The combination relies on the combined teachings of the references as understood by a POSITA and not on a physical incorporation of elements from one into another. *In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012); Ex.1003, ¶73.

### **5. Reasons to Combine Jeffries and LaBine with Ishaya**

A POSITA would have further combined Jeffries and LaBine with Ishaya because both Jeffries and Ishaya describe improving computer security by isolating containers. *See* Ex.1005, Abstract, 2:42-49 (“protect[ing] the computing device from attacks by isolat[ion]”); Ex.1007, [0001], [0194] (“increased security” through “identity-based authentication” of containers); Ex.1003, ¶¶74, 86.

The combination of Jeffries, LaBine, and Ishaya would include implementing the container authentication technique described in Ishaya in the combined Jeffries-LaBine system, such as within the management and monitoring service 104. Ex.1003, ¶¶75-76, 87.

Jeffries generally describes the importance of authentication. Ex.1005, 6:53-

57 (“requiring authentication from host operating system 102 before granting access to the one or more resources.”), 14:47-54 (user authentication). Jeffries mentions the process of “creat[ing] and activat[ing]” a new container. Ex.1005, 15:14-16. However, Jeffries combined with LaBine provides limited details regarding the identity of newly created containers or when the container is permitted to communicate after creation. Ex.1003, ¶¶77, 88.

Ishaya provides these implementation details. Given Jeffries’ limited details regarding container creation, a POSITA seeking to implement an isolated container as described by Jeffries would have found it obvious to consult Ishaya. Ishaya describes authenticating a container as part of this process by assigning it a MAC and/or IP address. Ex.1007, [0174]-[0186]; Ex.1003, ¶¶78-80. Ishaya further explains multiple advantages of its container creation process including authentication (e.g., ensuring that the container can be trusted, Ex.1007, [0189]), the ability to authenticate with controllers of other vendors, Ex.1007, [0190], as well as other advantages. *See* Ex.1007, [0191]-[0196]; Ex.1003, ¶¶78-79.

In Ishaya, a container is assigned an IP address which it uses to “authenticate itself.” Ex.1007, [0192], [0184]-[0185], [0187], Fig. 8; Ex.1003, ¶¶79-80.

Other advantages of Ishaya’s approach include “increas[ing]...flexibility and capability,” “increas[ing] security and privacy for users,” and “address[ing] the difficult problem of authentication during initial setup of a virtual machine

instance” by “remov[ing] the need for storing user credentials or encryption keys with a virtual machine image for authentication.” Ex.1007, [0193]-[0195]. Ishaya’s authentication technique also advantageously “provide[s] secure authentication and authorization of components within a multi-vendor cloud computing system.” Ex.1007, [0196]; Ex.1003, ¶¶81-82, 89.

Thus, the combination of Jeffries and LaBine further with Ishaya simply represents combining prior art elements according to known methods to yield predictable results. In the combination, a newly created isolated container, such as a container 122 described in Jeffries, *see* Ex.1005, 16:11-31, is authenticated as described by Ishaya, *see* Ex.1007, [0043], [0053], [0181]; *see also* Ex.1037, [0004], [0066] (“[T]rust characteristics attributable to components associated with a disaggregated infrastructure environment” such as “Docker containers or other types of containers.”). The results of the combination would have been predictable because Ishaya explains how this process of container isolation improves security with identity-based authentication as an improvement to simple user authentication with usernames and passwords. *See* Ex.1007, [0194]; Ex.1003, ¶¶83, 90.

In addition, a POSITA would have had a reasonable expectation of success of combining Jeffries and Labine with Ishaya. A POSITA would have sufficiently understood creating, isolating, and authenticating virtual machines and containers because these concepts were well known at the time. *See, e.g.*, Ex.1012, Abstract;

Ex.1013, Abstract; Ex.1014, [0076]; Ex.1015, [0011] (describing creating isolated computing environments); Ex.1018, [0003]; Ex.1019, [0018]; Ex.1020, [0180] (describing authenticating isolated computing environments). A POSITA would have expected to successfully combine Jeffries and Labine with Ishaya because both Jeffries and Ishaya involve isolating containers and virtual machines. *See* Ex.1005, 10:22-53, Abstract, Fig. 1, 2:39-60; Ex.1007, [0041]-[0043]. Both Jeffries and Ishaya describe the importance of authentication. *See* Ex.1005, 6:53-57, 14:47-54 (authenticating users); Ex.1007, Abstract, [0001] (authenticating containers). Authenticating containers in Jeffries' system would include assigning a MAC or IP address to the container after creation, and authenticating the container to another network device, as taught by Ishaya. In the combination, Jeffries' containers 122 would authenticate to the management and monitoring service 104 which "provides one or more administrative policies for the computing device" and/or to the web proxy 106 which authenticates "host operating system 102." Ex.1005, 6:44-57; Ex.1003, ¶¶84, 90.

For these reasons, a POSITA would have implemented the technique of authenticating containers, as taught by Ishaya, in Jeffries' system. Further, a container in Jeffries would include a video conferencing application, as described by LaBine. Doing so would predictably improve the security of the host computer through identity-based authentication as described by Ishaya. The combination

relies on the combined teachings of the references as understood by a POSITA and not on a physical incorporation of elements from one into another. *In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012); Ex.1003, ¶¶85, 91.

**6. Claim 1**

**[1.0] *A system comprising:***

Jeffries discloses a “*system*.” Jeffries’ Figure 1 “illustrates an example **system 100**.” Ex.1005, 6:20-22; Ex.1003, ¶¶92-94.

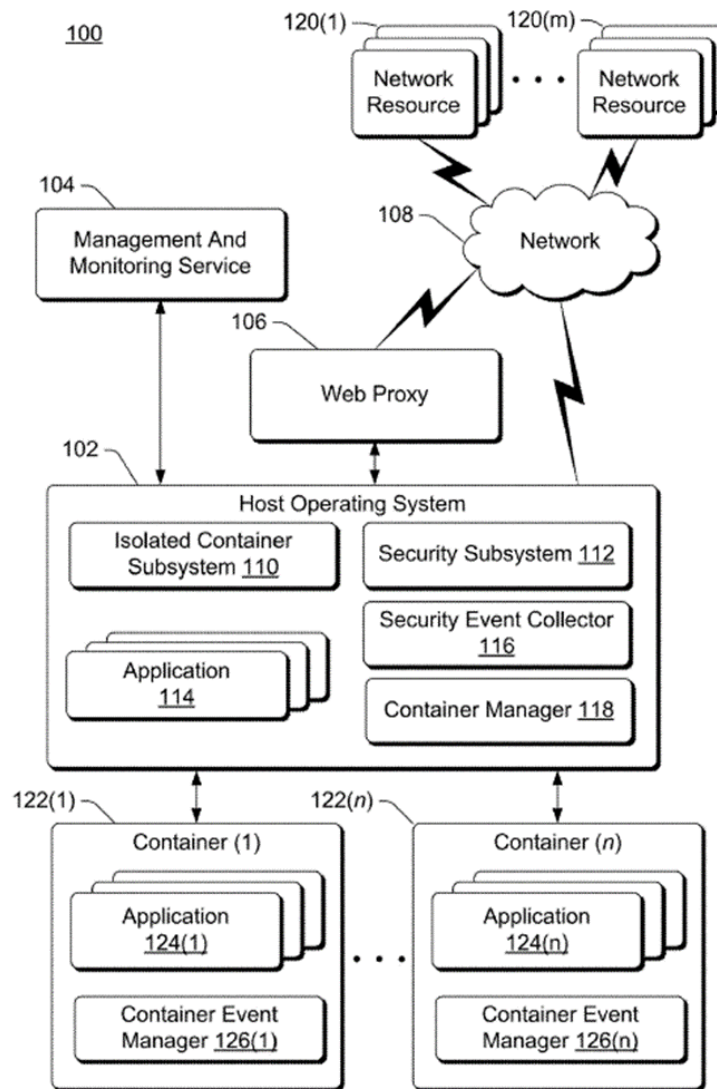


**Fig. 1**

Ex.1005, Fig. 1

[1.1] *a memory; and*

Jeffries discloses “*a memory.*” Ex.1003, ¶95.

Jeffries’ system 100 may be “a full resource device with **substantial memory**” or “a low-resource device with **limited memory.**” Ex.1005, 6:37-43;

Ex.1003, ¶96.

The “example computing device 602” in Figure 6 of Jeffries (below)  
“implement[s] the various techniques described herein.” Ex.1005, 22:39-42;  
Ex.1003, ¶97.

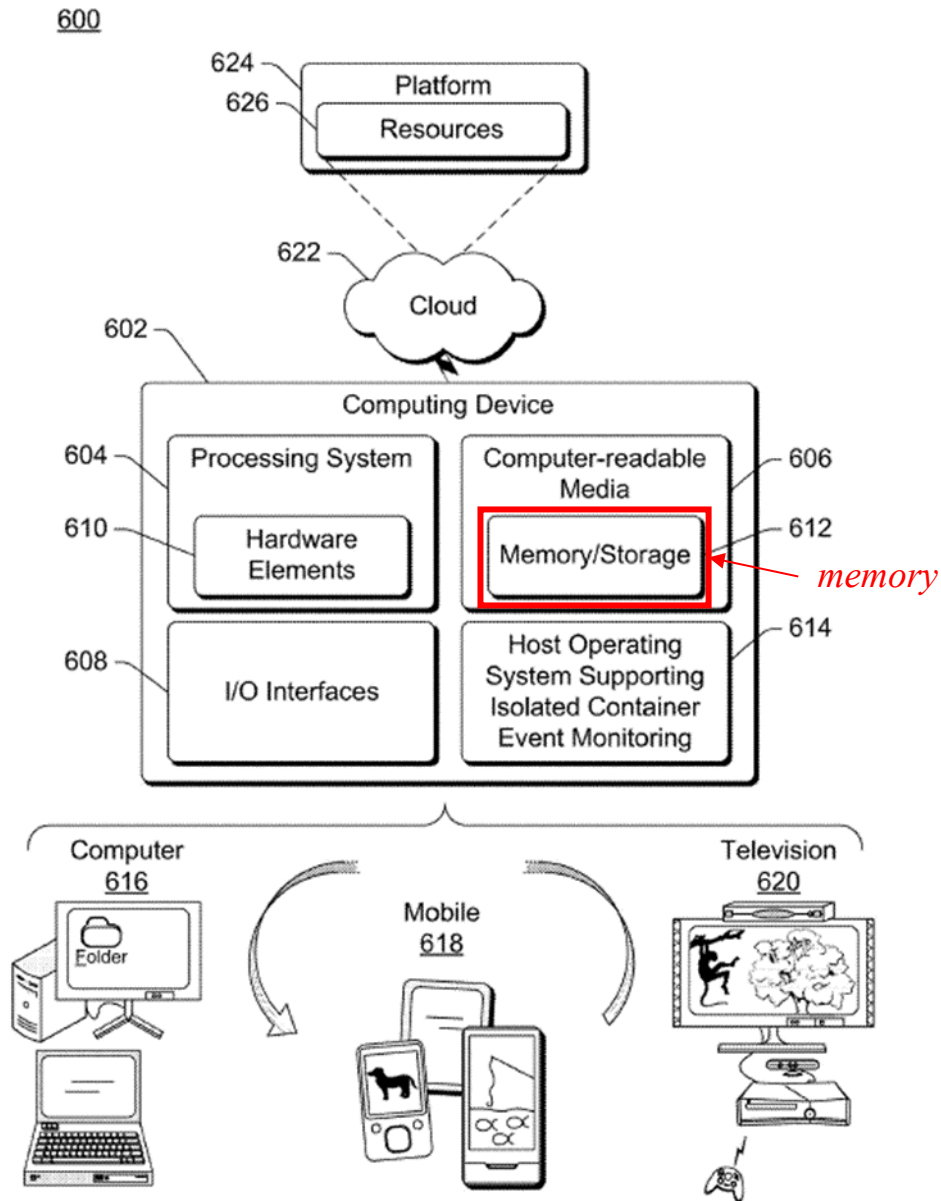


**Fig. 6**

Ex.1005, Fig. 6 (annotated)

As shown, the computing device 602 includes a “memory/storage 612,” which corresponds to “a memory” claimed. Ex.1005 23:8-10; Ex.1003, ¶¶98-99.

**[1.2] *a processor configured to:***

Jeffries' computing device 602 illustrated below also includes a “processing system 604” which corresponds to “*a processor*” claimed. Ex.1005, 22:47-50; Ex.1003, ¶¶100-101.

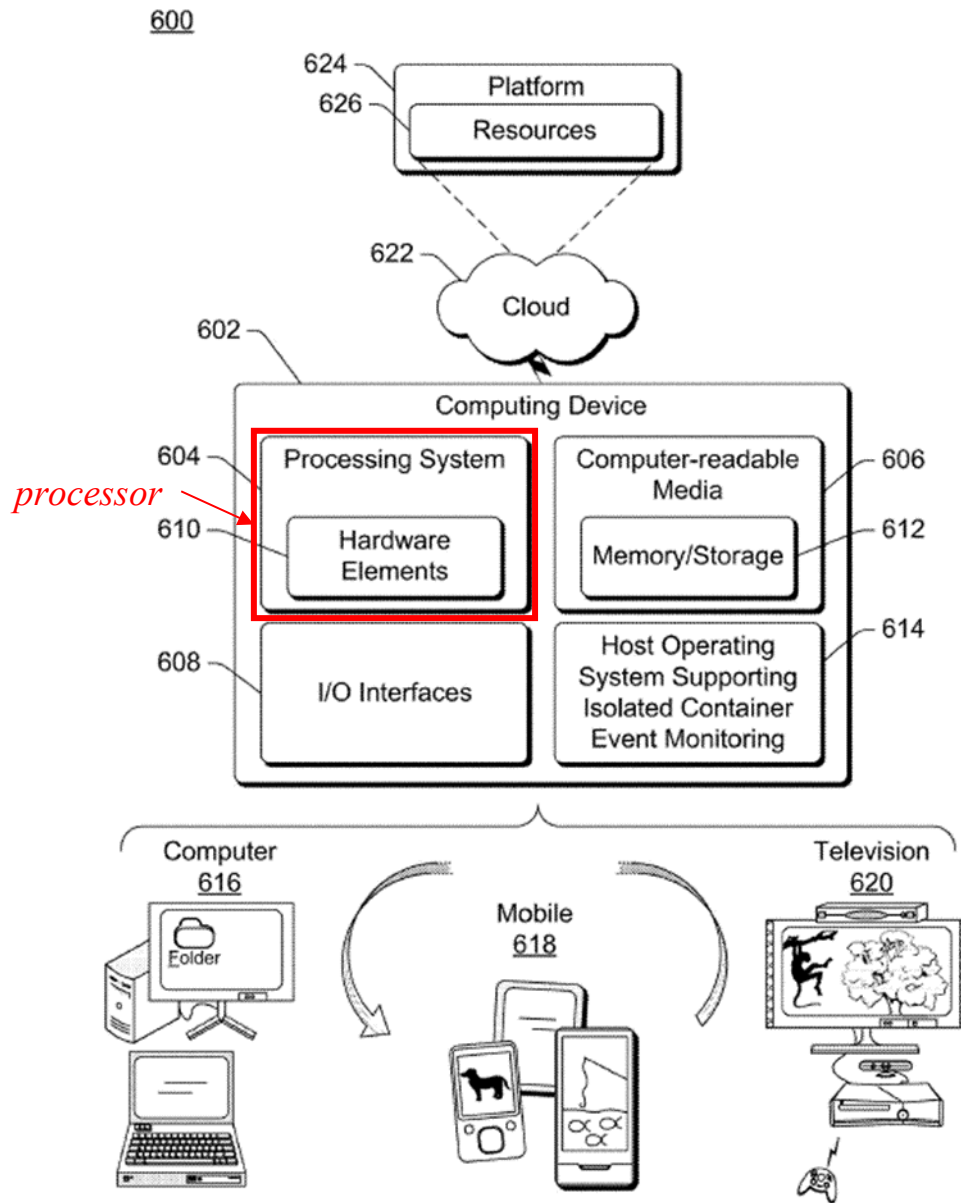


**Fig. 6**

**Ex.1005, Fig. 6 (annotated)**

Jeffries' processing system 604 "may be configured as **processors**."

Ex.1005, 22:60-64. Thus, Jeffries' processing system 604 corresponds to "a *processor*," as claimed. Ex.1003, ¶¶102-103.

**[1.3] *implement 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;***

Jeffries renders this limitation obvious. Ex.1003, ¶104.

**First**, Jeffries includes “*a host operating system.*” Jeffries’ “system 100 includes a host operating system 102,” highlighted in blue in Figure 1 below.

Ex.1005, 6:44-46; Ex.1003, ¶105.

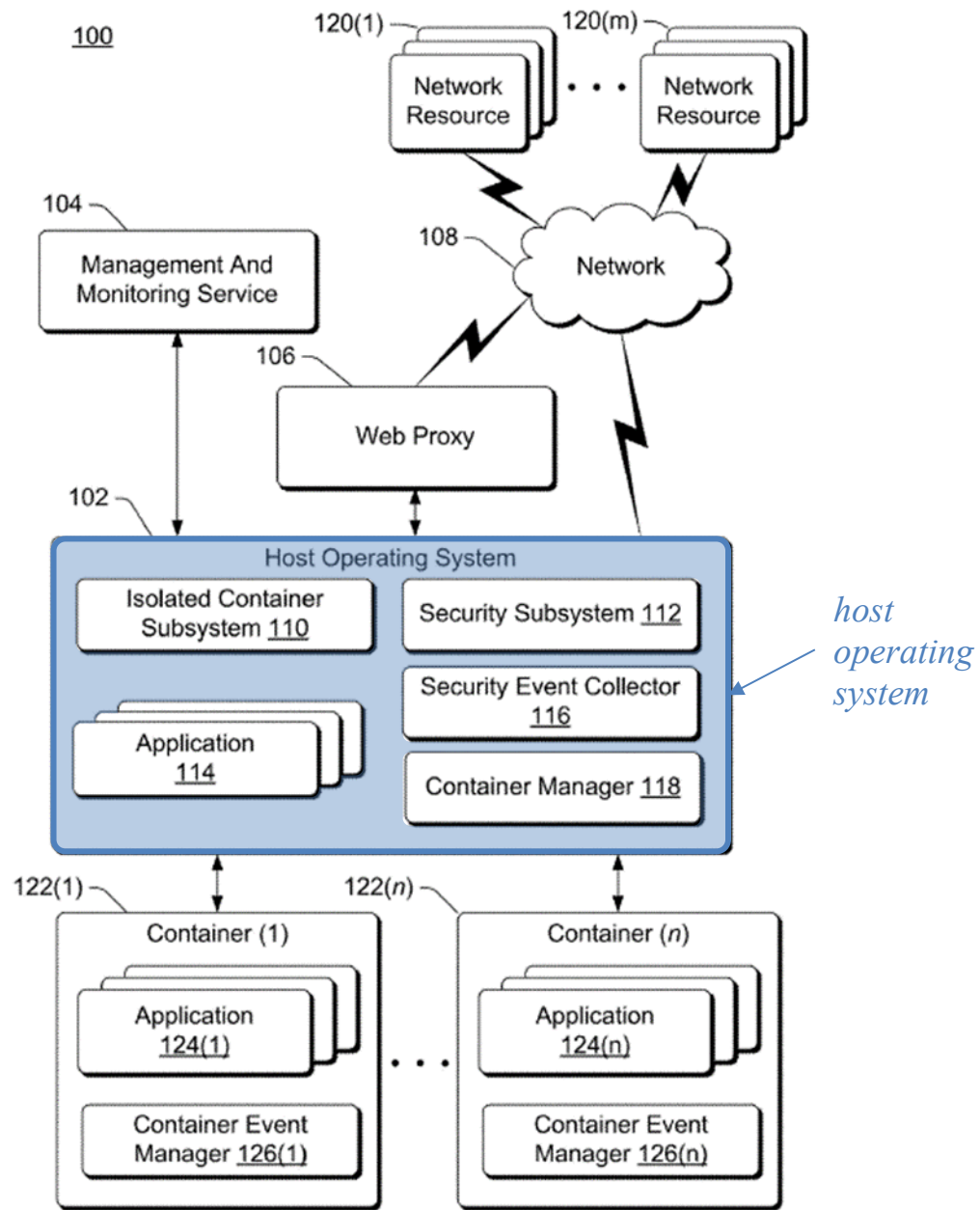


**Fig. 1**

**Ex.1005, Fig. 1 (annotated)**

**Second**, Jeffries' system "*implement[s] a workspace that uses*" the host operating system 102. Ex.1003, ¶106.

Jeffries describes resources that are not isolated within an isolated computing environment. These resources that are not isolated correspond to a “*workspace*.” For example, Figure 1 illustrates multiple components of the system 100 that are not isolated, highlighted in dark blue below. Ex.1003, ¶107.

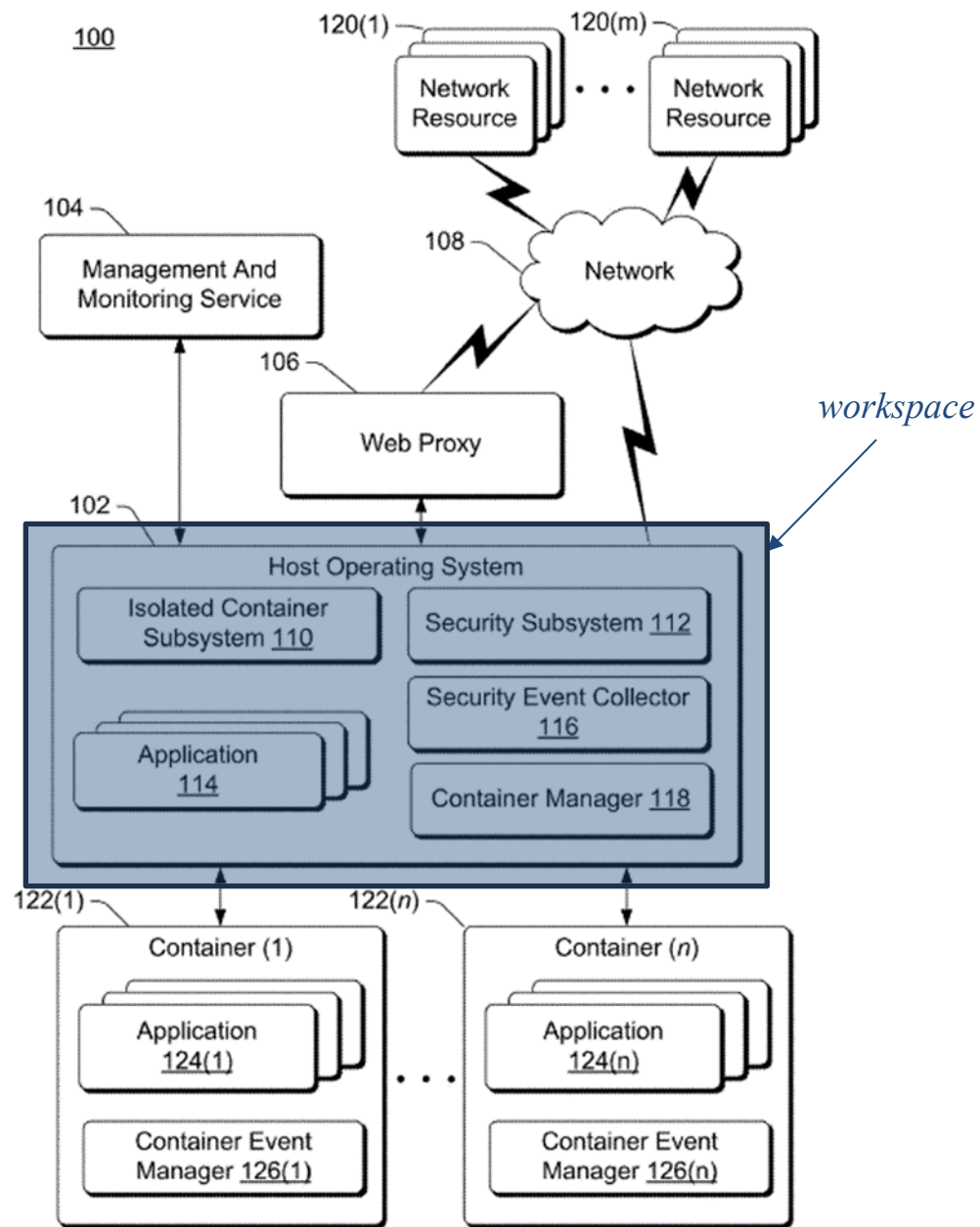


**Fig. 1**

**Ex.1005, Fig. 1 (annotated)**

The host operating system 102, including the isolated container subsystem 110, applications 114, security subsystem 112, security event collector 116, and

container manager 118, is not contained within the isolated computing environments (containers 122) in Jeffries and would therefore collectively be a “workspace,” as claimed and illustrated above. Ex.1003, ¶108.

The ’601 patent does not explicitly define the term “workspace,” but explains that it may include “one or more processes operating within the operating system that are not restricted by the sandbox container process.” Ex.1001, 5:44-46; *see also* Ex.1001, 1:65-2:2 (contrasting “**the isolated computing environment and the workspace.**”); Ex.1003, ¶108.

Alternatively, the “memory locations” supporting the components that are not isolated within isolated containers 122 in Jeffries may correspond to the claimed “workspace.” Ex.1005, 12:11-16. Specifically, Jeffries describes “memory locations associated with the container event manager 126” within isolated containers 122, as contrasted with memory locations associated with “the security event collector 116, the container manager 118, or **any other component of host operating system 102.**” Ex.1005, 12:11-16. The memory associated with the host operating system 102 is “off-limits” to the “container[s] 122.” Ex.1005, 12:7-22; Ex.1003, ¶¶109-110. Because these “off-limit data or memory locations” are not contained within Jeffries’ isolated computing environments, they may also correspond to a “workspace” as claimed. Ex.1003, ¶¶111-112; *see also* Ex.1005, 1:9-11 (describing “privileged memory locations”).

This alternative mapping of memory locations to the claimed “*workspace*” is also consistent with the ’601 patent. For example, the ’601 patent implies that the workspace is itself memory, including all memory that is not dedicated to the isolated computing environment. *See* Ex.1001, 5:33-37 (describing “a memory space that is isolated...from **another memory space** (e.g., **a workspace**)”); Ex.1003, ¶113.

Thus, Jeffries describes a “*workspace*” because it describes resources and memory locations that are not isolated within isolated containers 122. Ex.1003, ¶114.

**Third**, Jeffries’ resources and memory associated with the host operating system (the “*workspace*”) “*enables operation of a first set of one or more applications or processes.*” Ex.1003, ¶115.

Jeffries’ applications 114 within the host operating system 102 in Figure 1 are highlighted in green below. Ex.1003, ¶116.

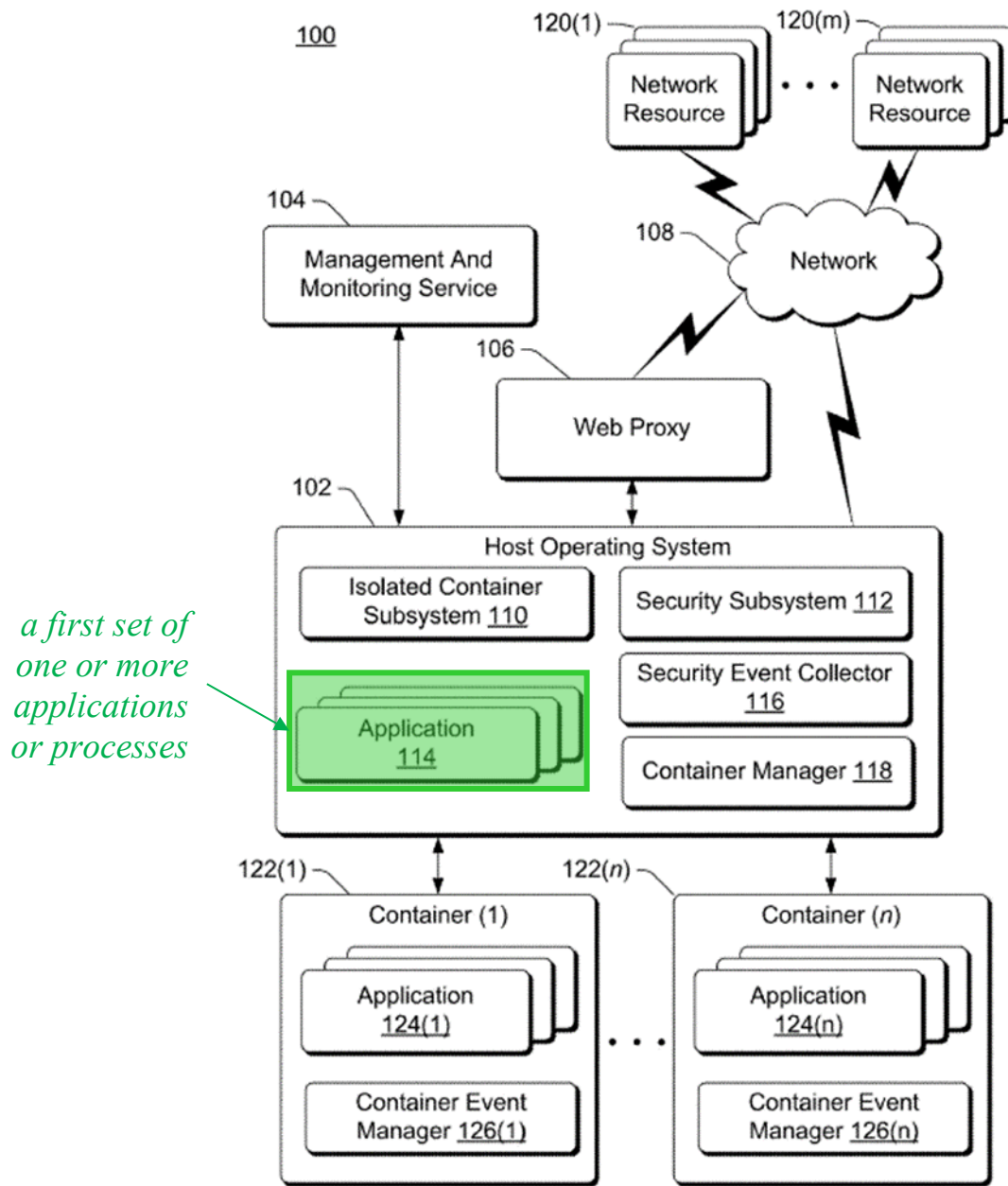


**Fig. 1**

**Ex.1005, Fig. 1 (annotated)**

The “[a]pplications 114 may include a web browser application...such as e-mail applications, word processing applications, spreadsheet applications, visual

presentation applications, and the like.” Ex.1005, 9:28-33. Because the applications 114 are executed within the host operating system 102, they are *enable[d]* by the workspace corresponding to the host operating system 102, including by general resources or actual memory not contained within the isolated containers 122. Ex.1003, ¶¶117-118.

**Fourth**, Jeffries’ applications 114 are enabled by the workspace “*via a first memory space*,” as claimed. Ex.1003, ¶119.

As explained above, Jeffries describes “memory locations” associated with the host operating system 102. Ex.1005, 12:12-16. Specifically, Jeffries describes “**memory locations associated with**” each component of the host operating system 102, including “the container event manager 126, the security event collector 116, the container manager 118, or **any other component of host operating system 102.**” Ex.1005, 12:11-16; Ex.1003, ¶120.

The applications 114 are a “component of the host operating system 102,” as illustrated in Figure 1. Ex.1005, 12:11-16. Therefore, Jeffries’ system includes “memory locations” associated with the applications 114. These memory locations correspond to “*a first memory space*” that enable the operation of the applications 114. Ex.1003, ¶121. A POSITA also would have understood that memory locations associated with applications 114 would enable operation of the applications 114 because this arrangement would have been consistent with basic principles of

computer architecture at the time. *See, e.g.*, Ex.1024, [0003]; Ex.1025, Abstract; Ex.1026, Abstract; Ex.1003, ¶122.

Thus, Jeffries describes resources and/or a memory space that is not isolated within isolated containers 122 (“*implement a workspace*”) and supports a host operating system 102 (“*that uses a host operating system*”). In addition, the resources/memory space not isolated within the isolated containers 122 (“*wherein the workspace*”) supports applications 114 (“*enables operation of a first set of one or more applications or processes*”) with memory locations associated with the applications 114 (“*via a first memory space*”). Ex.1003, ¶123.

**[1.4.1] *implement an isolated computing environment that uses the host operating system, wherein the isolated computing environment enables operation of a second set of one or more applications or processes via a second memory space, and***

Jeffries renders this limitation obvious. Ex.1003, ¶124.

**First**, Jeffries describes “*implement[ing] an isolated computing environment.*” Ex.1003, ¶125. Jeffries’ “containers 122(1),..., 122(n)” are “*isolated computing environment[s]*.” Ex.1005, 8:14-17. Jeffries’ containers 122 are illustrated in Figure 1, with container 122(1) highlighted in orange. Ex.1003, ¶126.

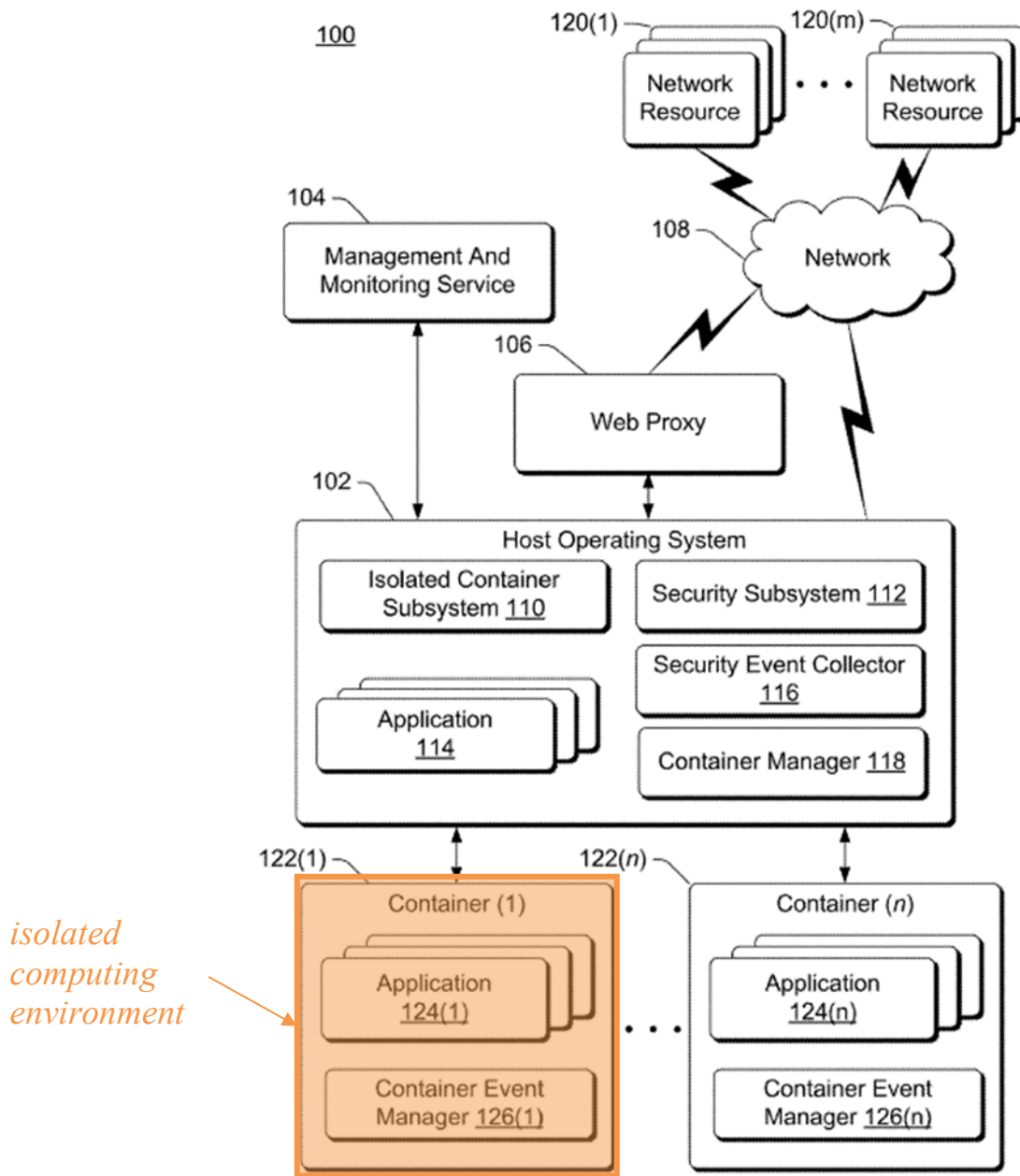


**Fig. 1**

**Ex.1005, Fig. 1 (annotated)**

Jeffries containers 122 are described as “isolated containers.” Ex.1005, 2:39-40. Isolated containers 122 “provide full kernel isolation from the host operating

system.” Ex.1005, 2:52-55. Each container 122 is “isolated from the host operating system” to “prevent...malicious activity from spreading beyond the isolated container.” Ex.1005, Abstract; Ex.1003, ¶127.

**Second**, Jeffries describes how its isolated container 122 (an “*isolated computing environment*”) “*uses the host operating system*.” Ex.1003, ¶128.

In Jeffries, processes within an isolated container 122 access “**the shared parts of host operating system 102.**” Ex.1005, 10:25-34; Ex.1003, ¶129.

Accessing “shared parts of host operating system 102” corresponds to “*us[ing] the host operating system,*” as claimed. Ex.1005, 10:29-34; Ex.1003, ¶130.

In addition, Jeffries uses the same method of isolating containers as the ’601 patent: namespace isolation. The ’601 patent describes “enforc[ing] the segregation using techniques such as namespace isolation” to allow “the sandboxed computing environment to share access to the host operating system and host operating resources.” Ex.1001, 5:57-63. In Jeffries, the “[h]ost operation system 102 implements namespace isolation.” Ex.1005, 10:28-29. Therefore, Jeffries’ isolated containers 122 use the host operating system 102 just as the ’601 patent’s “sandboxed computing environment” uses the host operating system. Ex.1001, 5:64-63; Ex.1003, ¶¶131-132.

During prosecution, the patent owner added the phrase “*uses the host operating system*” to overcome an office action citing Azab. The patent owner

argued that Azab “does not rely on any software component in the host environment (i.e., an OS or a hypervisor).” Ex.1002, 217 (quoting Ex.1034, Abstract); Ex.1003, ¶133. Unlike Azab, Jeffries’s containers 122 rely on several software components in the host operating system 102, including container manager 118.

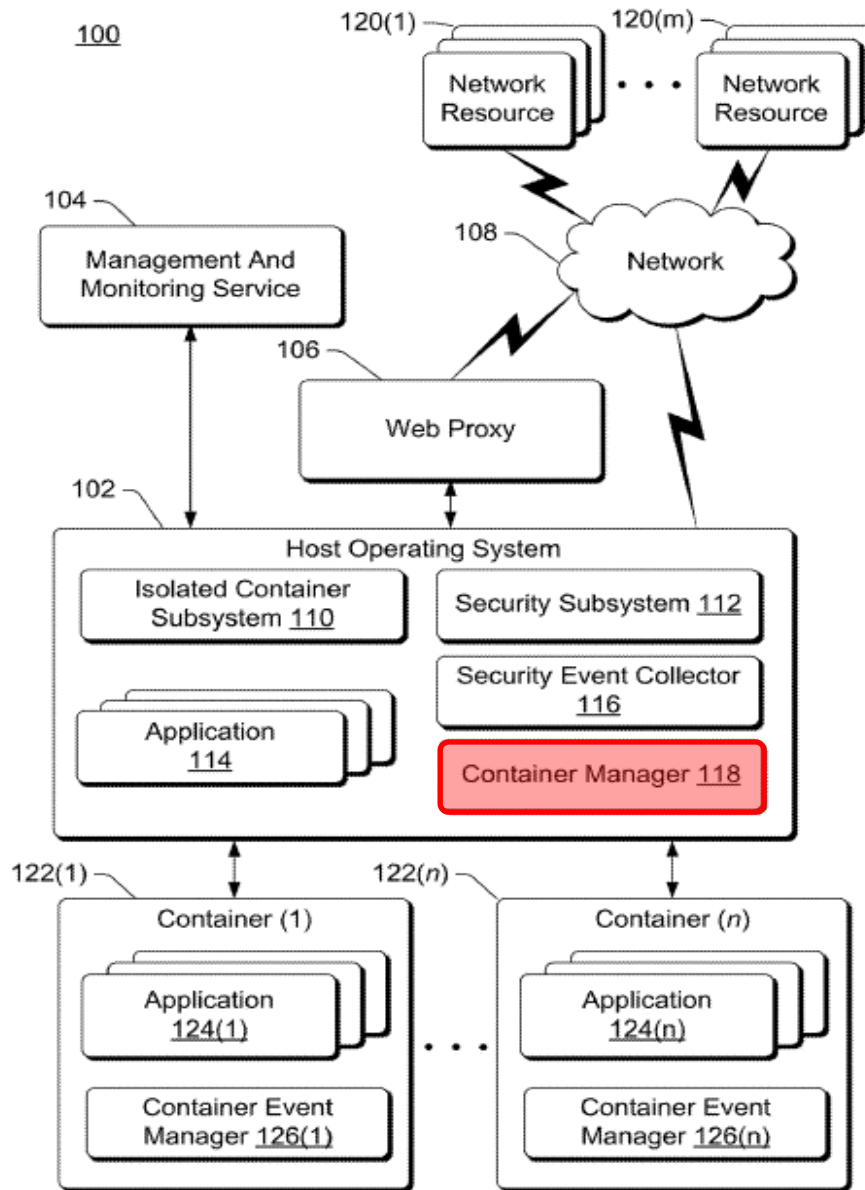


**Fig. 1**

Jeffries, Fig. 1 (annotated)

The container manager 118 “manages the scheduling of containers 122(1),..., 122(n) in the system 100 and determines which containers 122(1),..., 122(n) are run on the host operating system 102 at what times.” Ex.1005, 9:62-67;

Ex.1003, ¶¶134-135. In addition, the isolated container subsystem 110 “causes the host operating system 102 to activate one or more of containers 122(1),..., 122(n) and allow the one or more activated containers to access the untrusted network resource.” Ex.1005, 8:67-9:6. The “security subsystem 112” and “security event collector 116” monitor “container activity resulting from resource access in one or more of containers 122(1),..., 122(n).” Ex.1005, 9:7-15; Ex.1003, ¶136. Thus, Jeffries’ containers 122 use the host operating system for both container monitoring and initial activation and “run on the host operating system 102” while protecting the host operating system 102 from malware through isolation. Ex.1005, 9:62-67; Ex.1003, ¶137.

**Third**, Jeffries describes “*the isolated computing environment enables operation of a second set of one or more applications or processes via a second memory space.*” Ex.1003, ¶138.

Jeffries illustrates a set of applications 124(1) running in isolated container 122(1), as highlighted in purple below. Ex.1003, ¶¶139-140.

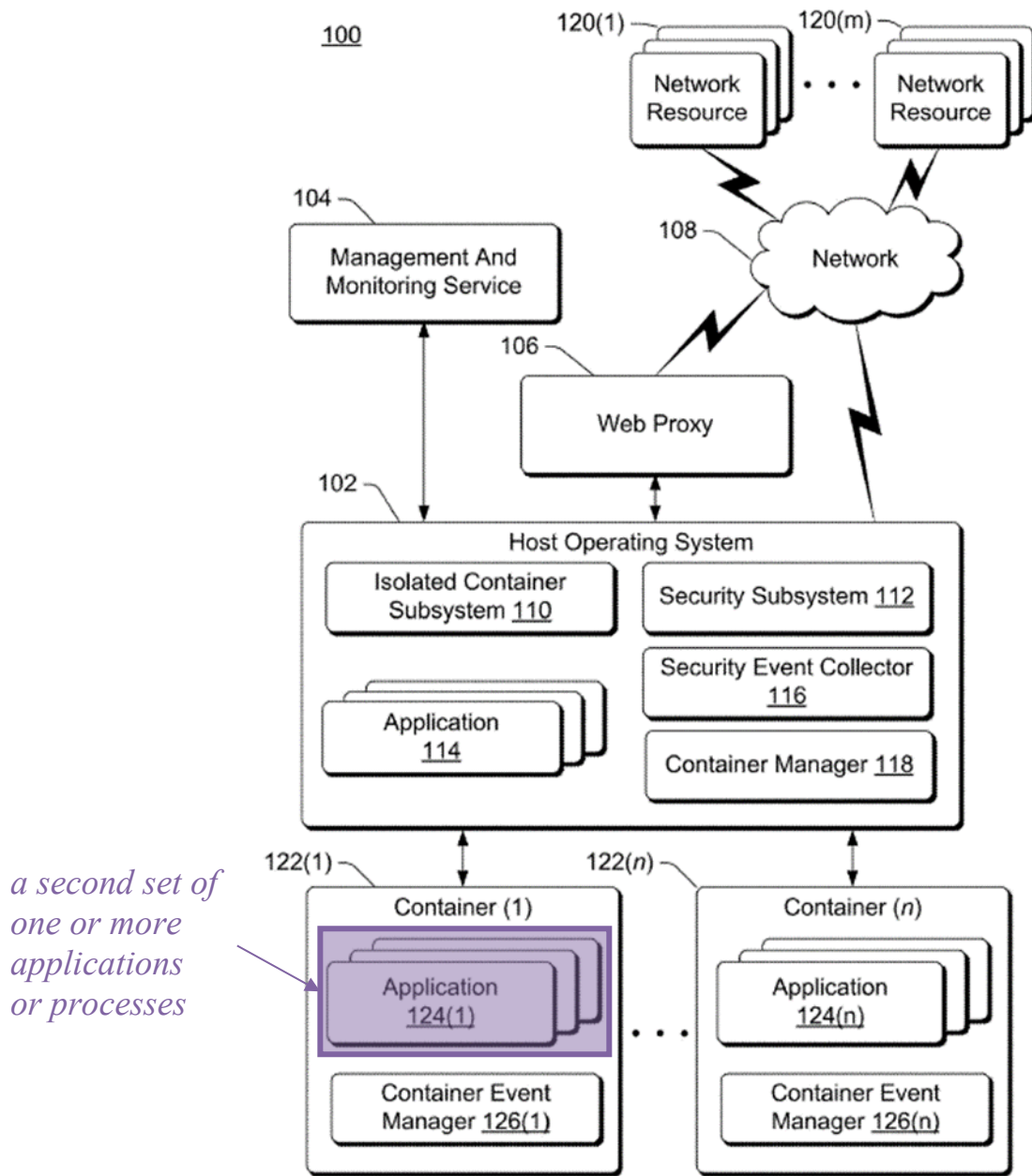


**Fig. 1**

**Ex.1005, Fig. 1 (annotated)**

The applications 124 within containers 122 are associated with separate memory space isolated from the memory space of the host operating system 102,

including applications 114. Jeffries describes “isolated parts of the operating system that are specific to each container” which “may also be referred to as container subsystems.” Ex.1005, 10:34-37, Fig. 5 (“Run An Application In A Container That Is Isolated From A Host Operating System”); Ex.1003, ¶141.

Jeffries also explains that “[a] container base image for each container 122 “includes information regarding an amount of allocated resources that may be required by container subsystems 206, **such as memory**, processors, disks, or networks, when container 122 is active.” Ex.1005, 15:14-30. In Jeffries, containers 122 are associated with “**container memory**.” See Ex.1005, 17:7-20, 21:4-8. Container base image memory or container memory correspond to “*a second memory space*” for the operation of applications 124 within a container 122. Ex.1003, ¶¶142-144.

Thus, Jeffries describes creating and isolating containers 122 (“*implement[ing] an isolated computing environment*”) that access portions of the host operating system 102 and use the host operating system 102 for monitoring and activation (“*that uses the host operating system*”). Jeffries also describes a set of applications 124 within the isolated containers 122 (“*wherein the isolated computing environment enables operation of a second set of one or more applications or processes*”) and explains that container memory is allocated to these applications 124 (“*via a second memory space*”) as claimed. Ex.1003, ¶145.

**[1.4.2] *wherein the second set of one or more applications or processes comprises a multi-user interactive software application;***

The combination of Jeffries and LaBine renders this limitation obvious.

Ex.1003, ¶146.

As described above, *see* limitation [1.4.1], Jeffries describes applications 124 in containers 122 that “access untrusted resources” Ex.1005, 10:58-63; Ex.1003, ¶147. Jeffries does not list every type of application 124 that may be executed in the isolated containers 122 in detail. Rather, it provides a non-limiting list of broad categories. *See* Ex.1005, 9:28-33 (“a web browser application,” “e-mail applications, word processing applications, spreadsheet applications, visual presentation applications, and the like”); *see also* Ex.1005, 10:54-58 (applications 124 “correspond to instances of individual ones of the applications 114.”). Jeffries’ techniques of executing applications within isolated containers would have been applied to any type of application. Ex.1003, ¶148; *see supra* §X.B.4.

LaBine describes a “*multi-user interactive software application.*” LaBine describes a video conferencing software that allows participants of the video conference to remotely access and control the host user’s computer. *See* Ex.1006, Abstract, [0078]-[0084]; Ex.1003, ¶149.

LaBine’s software is referred to as the “SMART Bridgit Conferencing Software,” and is video conferencing software implemented in the computing

environment 20 shown below. Ex.1006, [0043], [0045]; Ex.1003, ¶150.

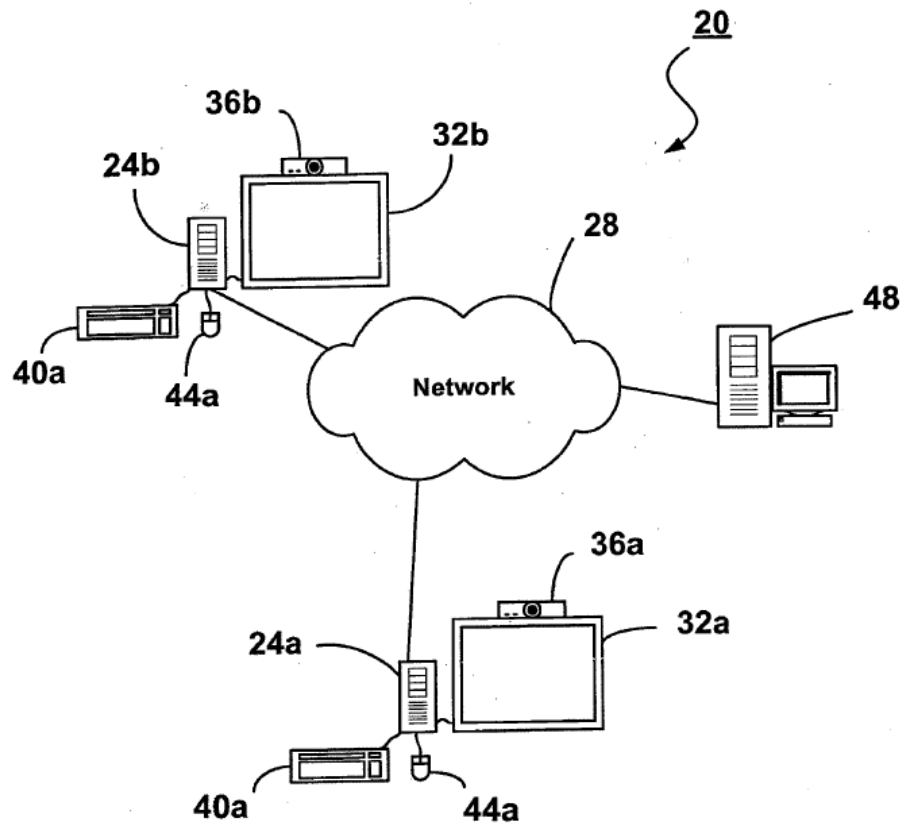


**Fig. 1**

**Ex.1006, Fig. 1**

The “Bridgit Conferencing Software...allows a conference to be set up between personal computers with one personal computer at a time being designated as a host, or ‘presenter’ computer and the remaining personal computers being designated as client, or ‘participant’ computers.” Ex.1006, [0045]. “The client computers can be enabled to control the host computer.” Ex.1006,

[0045]; Ex.1003, ¶151.

During a conferencing session, a client computer sends a “REMOTE CONTROL request.” Ex.1006, [0078]. If the “host computer...approves the request,” it sends “a REMOTE CONTROL RESPONSE” which grants the client computer remote control of the host computer. Ex.1006, [0080]-[0081]. With remote control of the host computer, “remote user input events on the client computer..., such as mouse, keyboard and pen events, are received at the conferencing server 48 from the remote controller and routed to the host computer.” Ex.1006, [0084]. The host computer “treats the remote user input events as though they were input locally.” Ex.1006, [0084]; Ex.1003, ¶152.

The graphical user interface displayed to a remote-control user of LaBine’s system is shown in Figures 22-24 below. Ex.1003, ¶¶153-155.

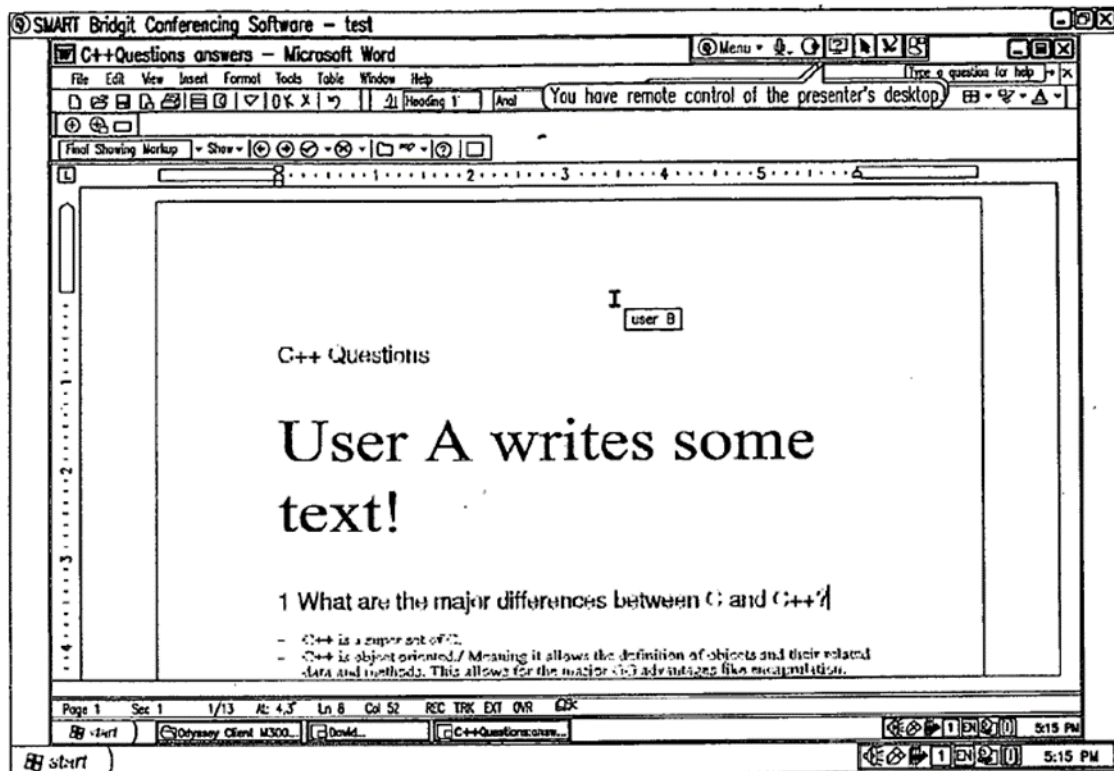


**FIG. 22**

Ex.1006, Fig. 22

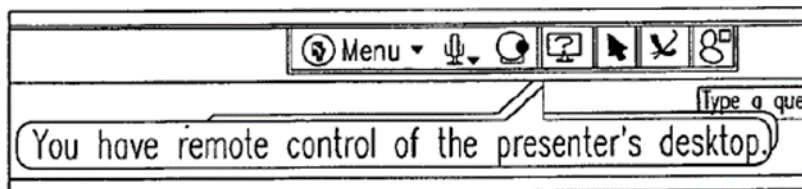


**FIG. 23**

Ex.1006, Fig. 23

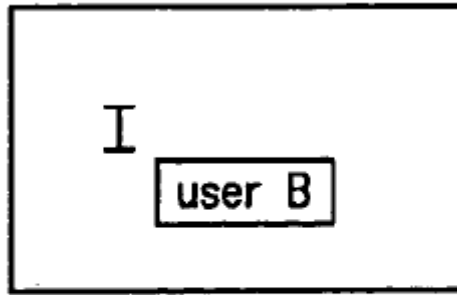


**FIG. 24**

Ex.1006, Fig. 24

LaBine’s video conferencing software is a “*multi-user interactive software application*” as defined by the ’601 patent because it is “an application that allows one or more remote machines to control (e.g., access, view, interact with, download from, upload to, and/or modify) an application running on the host computer system.” Ex.1001, 2:18-23; *see also* §X.B.7, [2.1]-[2.2]; Ex.1003, ¶156.

A POSITA would have been motivated to implement a video conferencing and remote control application as described by LaBine within an isolated container as described by Jeffries. *See supra*, §X.B.4. In the combination, LaBine’s application would be executed within Jeffries’ isolated container. Ex.1003, ¶157.

Thus, the combination of Jeffries and LaBine describes a software application within an isolated container (“*the second set of one or more applications or processes*”) that allows interactions with other users, including

allowing a remote user to access and control portions of the host computer (“a multi-user interactive software application”). Ex.1003, ¶158.

**[1.5] isolate the isolated computing environment from the workspace using an internal isolation firewall;**

Jeffries’ container manager 118, in conjunction with the operating system secure kernel, renders this limitation obvious. The container manager 118 is highlighted in dark green in Figure 2 below. Ex.1003, ¶¶159-160.

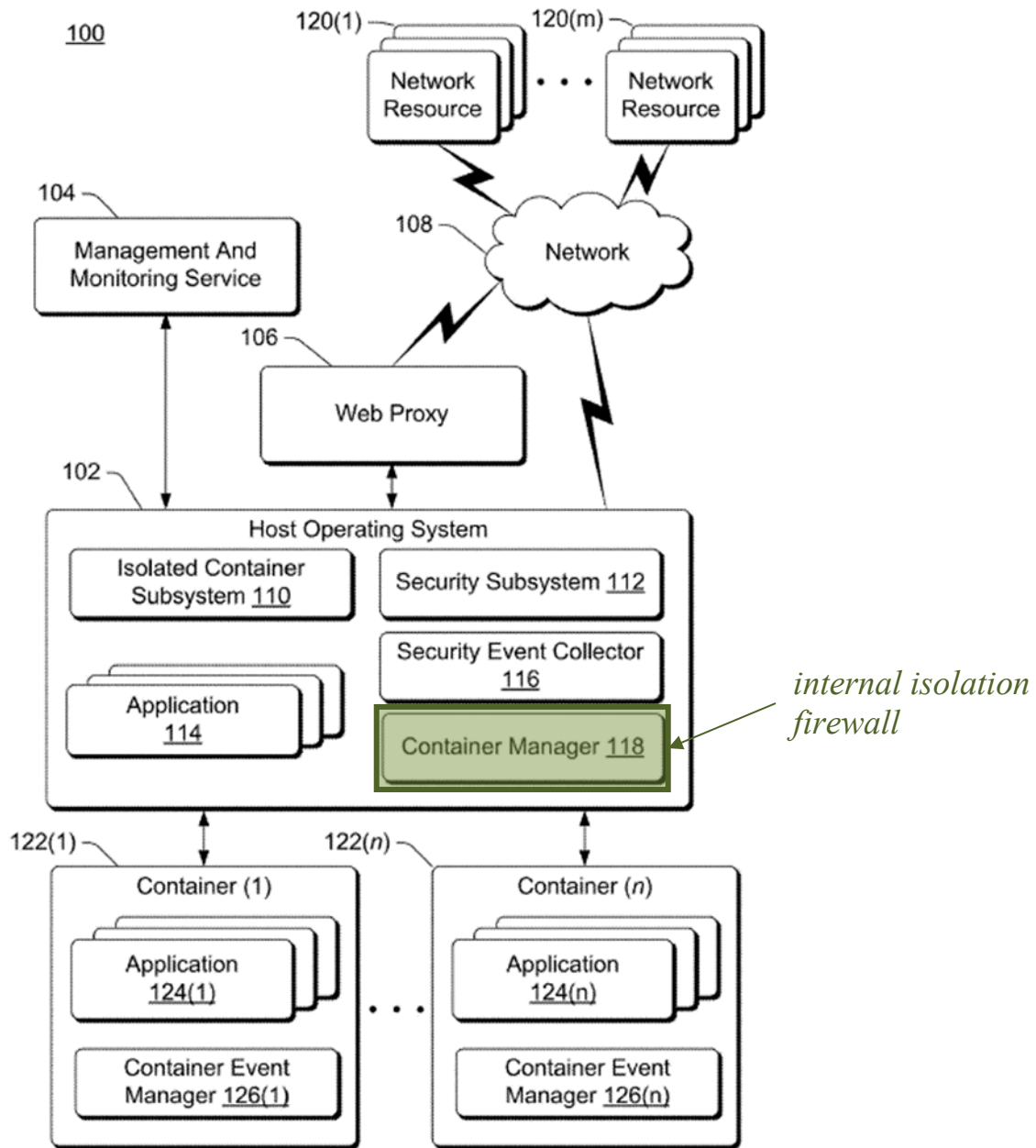


**Fig. 1**

Ex.1005, Fig. 1 (annotated)

The container manager 118 acts as an “internal isolation firewall” because it “is configured to **quarantine the container 122**” and “**prevents** malicious activity

occurring in the container from gaining... **access to shared resources.**” Ex.1005, 16:24-37; *see also id.*, 16:38-46, 14:40-44 (“take corrective action in response to... malicious activity”). The secure kernel (not pictured) helps to enforce the separation between the host operating system 102 and the container 122. *See* Ex.1005, 12:7-22 (“[W]hen malicious software attempts to access off-limit data or memory locations, the secure kernel causes an exception or interrupt to occur.”); Ex.1003, ¶¶160-161.

The container 118 and secure kernel described in Jeffries correspond to an “*internal isolation firewall*” as claimed because they perform the same functions of the internal isolation firewall described by the ’601 patent, including preventing communication between the isolated computing environment and the workspace. *See, e.g.*, Ex.1001, 1:67-2:2, 3:38-40, 6:63-66, 13:15-23, 14:7-14; Ex.1003, ¶¶162-163.

In summary, Jeffries describes isolating the containers 122 from the remainder of the host operating system 102 (“*isolat[ing] the isolated computing environment from the workspace*”) with a container manager 118 in conjunction with the secure kernel (“*using an internal isolation firewall*”). Ex.1003, ¶164.

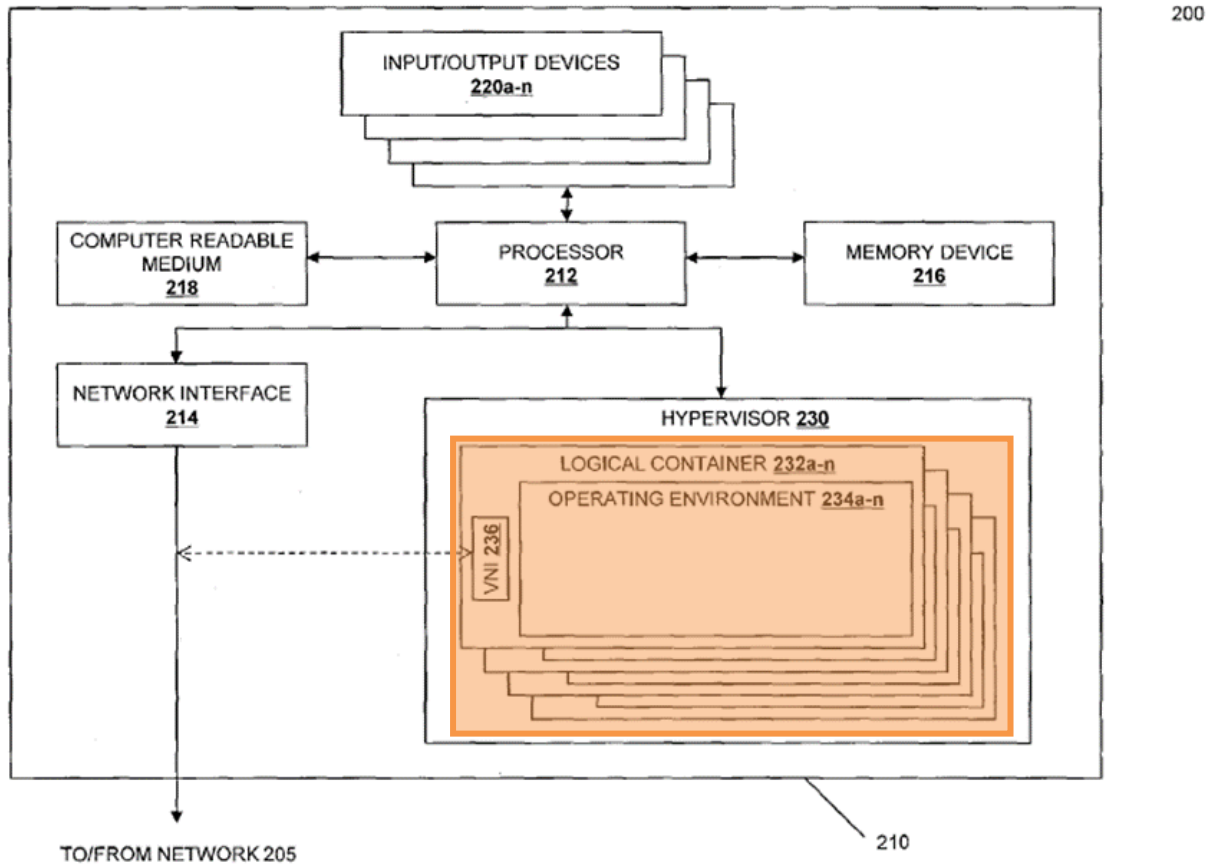
**[1.6] *authenticate the isolated computing environment with an authentication device; and***

The combination of Jeffries, LaBine, and Ishaya renders this limitation

obvious. Ex.1003, ¶165.

Jeffries describes authenticating the host operating system 102 and users of isolated containers 122. *See, e.g.*, Ex.1005, 6:53-57 (“requiring authentication from host operating system 102”), 14:47-52, 14:56-69 (describing authenticating users), 15:9-13, 15:23-26; Ex.1003, ¶166.

Ishaya describes authenticating isolated computing environments, referred to as “containers.” Ex.1007, [0041]. Figure 2, below, illustrates “a personal computer,” Ex.1007, [0035], that includes “a number of logical containers 232a-n” (highlighted in orange below). Ex.1007, [0041]; Ex.1003, ¶167.



**Ex.1007, Fig. 2 (annotated)**

Like the '601 patent and Jeffries, the containers 232 in Ishaya may be isolated from the remainder of the system. *See* Ex.1007, [0041] (“A corresponding operating environment 234 could enforce separation between users and processes such that each process or group of processes appeared to have sole access to the resources of the operating system.”); Ex.1003, ¶168.

Ishaya explains the process by which a container is instantiated with reference to Figure 7, shown below. Ex.1003, ¶169.

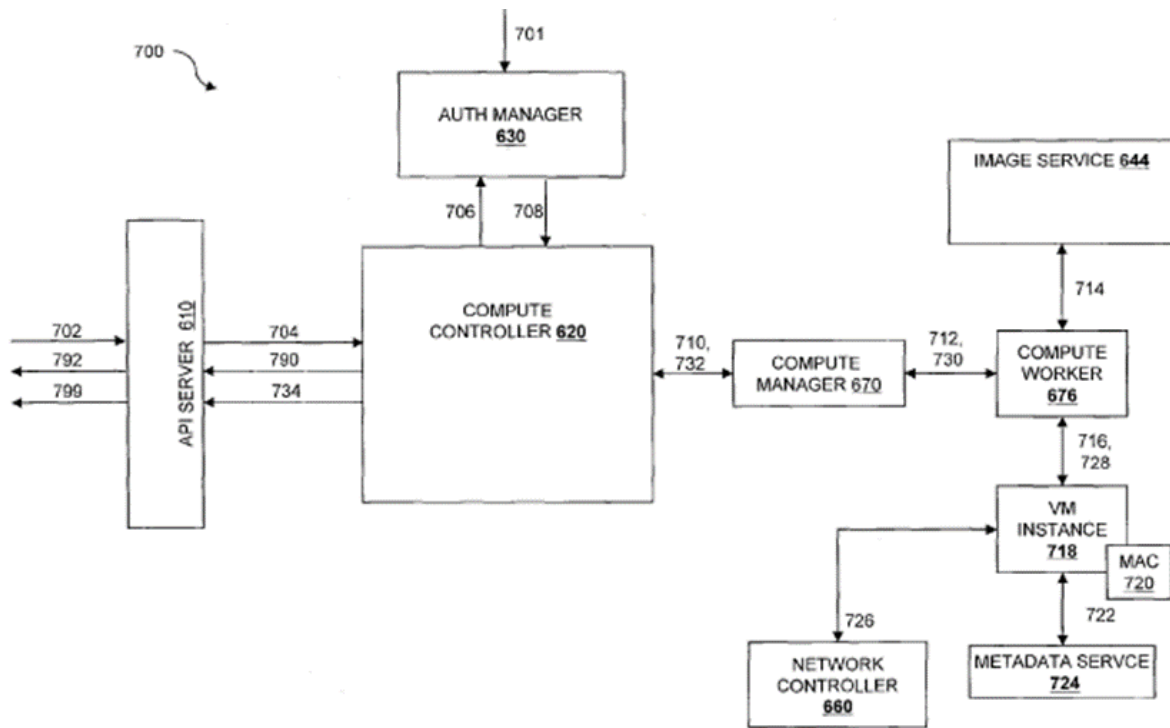


FIGURE 7

**Ex.1007, Fig. 7**

After a container is instantiated, “the Computer Worker assigns the VM instance 718 a MAC address 720” which is “selected from a pool of MAC addresses.” Ex.1007, [0181]. The MAC address may be “replaced with...an IP address.” Ex.1007, [0181]. Then, “[a]t time 722, the newly instantiated VM instance 718...authenticates to the Metadata Service 724 via an identity” such as its MAC address or IP address. Ex.1007, [0183]-[0184]. After this initial authentication, the VM instance receives “possession-based credentials” that it can use “for subsequent authentication to other resources.” Ex.1007, [0184]. For example, at time 726, “[t]he VM instance 718 again provides its MAC address 720

in order to authenticate itself to the Network Controller 660.” Ex.1007, [0185]; Ex.1003, ¶170.

Ishaya describes multiple benefits of authenticating containers.

Authenticating an isolated container allows different devices, including other virtual machines or containers to “determine whether or not to trust information from [the container] based on the identity value of” the container. Ex.1007, [0189]. In addition, it increases security even while interoperating with different devices by allowing “controllers from different vendors or entities” to “authenticate to each other.” Ex.1007, [0190]. Ishaya lists multiple other advantages stemming from container authentication. *See also* Ex.1007, [0193]-[0196] (discussing further advantages); Ex.1003, ¶171.

For these reasons, *see* §X.B.5, a POSITA would have been motivated to authenticate containers 122 with an authentication device, such as a device similar to the Network Controller 660 described in Ishaya, using a unique device identifier, such as a MAC address or IP address. In the combination, Jeffries’ isolated container 122 would be instantiated using the steps described in Ishaya, *see* Ex.1007, [0177]-[0187], and assigned a MAC or IP address, after which the container 122 would be authenticated to an authentication device, such as a

metadata service or network controller.<sup>2</sup> Ex.1003, ¶172.

Therefore, the combination of Jeffries and Ishaya describes authenticating an isolated container to a network controller (*authenticat[ing] the isolated computing environment with an authentication device*). Ex.1003, ¶174.

**[1.7] 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.**

The combination of Jeffries, LaBine, and Ishaya renders this limitation obvious. Ex.1003, ¶175.

In the combined system, a video conferencing application, such as the one described by LaBine, is executed within an isolated container, as described in Jeffries, after the container is authenticated, as described by Ishaya. Ex.1003, ¶176.

LaBine's multi-user interactive software, *see* §X.B.6 (limitation [1.4.2]), *send[s] data to an untrusted destination* because it allows participants of the video

---

<sup>2</sup> The functions of assigning a MAC or IP address to a container 122 and authenticating the container 122 may be performed by the management and monitoring service 104, web proxy 106, security subsystem 112, or isolated container subsystem 110 of Jeffries' system, or an additional server or network device dedicated to authentication of isolated containers. Ex.1003, ¶173.

conference to share a view of their computer screen. Ex.1006, [0051], Figs. 22-23. The users of the client computers in Labine may be any user, known or unknown. But regardless of the level of trust of the user of the client computer, LaBine allows “**any** client computer during the conference to be granted remote control of the host computer.” Ex.1006, [0075]. Allowing any client computer to control the host computer would also involve sending data to an untrusted destination. For example, the “REMOTE CONTROL RESPONSE” sent from the host to the client requesting remote control would correspond to data sent to an untrusted destination. *See* Ex.1006, [0046], [0048], [0050]; Ex.1003, ¶177. Similarly, Jeffries’ “applications 124(1),..., 124(n) are thus useable to access **untrusted resources** in one or more of containers 122(1),..., 122(n).” Ex.1005, 10:58-63, 3:48-51; Ex.1003, ¶178.

In Jeffries, communication with an untrusted destination occurs via the web proxy 106, which corresponds to “*a proxy device*.” The web proxy 106 is positioned between the host computer (including the host operating system 102 and the isolated containers 122) and external network resources 120, as shown in Figure 1 below, highlighted in dark red. Ex.1003, ¶179.

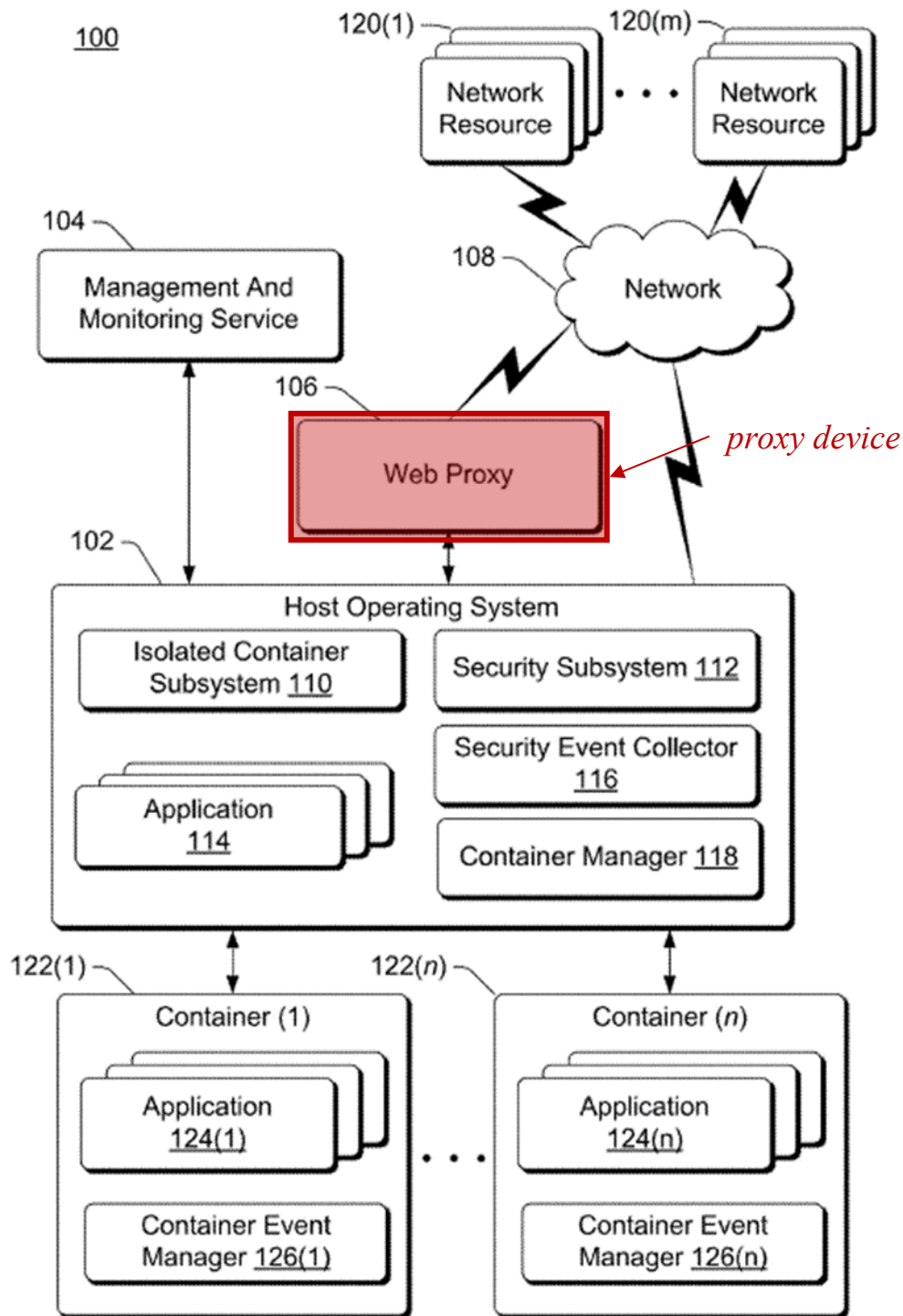


**Fig. 1**

Ex.1005, Fig. 1 (annotated)

The web proxy 106 “controls access to one or more network resources 120(1),..., 120(m).” Ex.1005, 6:49-53. In one embodiment, the network resources 120 represent external network resources, such as “web sites, web applications, emails, [or] documents,” Ex.1005, 6:64-67, that are treated as untrusted resources and “access[ed]... in a container 122(1),..., 122(n),” Ex.1005, 7:50-54; Ex.1003, ¶180.

In the proposed combination, the isolated container 122 in Jeffries would be authenticated, as described by Ishaya. *See* §X.B.6 (limitation [1.6]). Ishaya’s isolated container is not permitted to communicate with any source until after it is authenticated. *See* Ex.1007, [0183]. At step 820, the “VM uses the MAC address to authenticate itself to various configuration resources.” Ex.1007, Fig. 8, [0192]. After this authentication step, at step 822, the final step, the “VM is configured and ready for regular use.” Ex.1007, Fig. 8, [0192]. In the combination, an application, such as LaBine’s collaboration application, runs within the isolated and authenticated container and accesses untrusted sources. *See* §X.B.6 (limitation [1.4.1]). It would have been obvious from Ishaya for LaBine’s collaboration application within a container to run once the container is “configured and ready,” and hence, after the container is authenticated. Ex.1003, ¶181.

Thus, the combination describes transmitting data from a video conferencing application to untrusted destinations, as described by Jeffries and LaBine (*send*

*data to an untrusted destination from the multi-user interactive software application*), through the web proxy 106, as described by Jeffries (*via a proxy device*), after the container 122 has been authenticated, as described by Ishaya (*when the isolated computing environment has been authenticated*). Ex.1003, ¶182.

## 7. Claim 2

### **[2.0] *The system of claim 1,***

The combination of Jeffries, LaBine, and Ishaya renders obvious *the system of claim 1*. Ex.1003, ¶183.

### **[2.1] *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,***

The combination of Jeffries and LaBine renders this limitation obvious. Ex.1003, ¶184.

In the combined system, the video conferencing application described by LaBine, *see* Ex.1006, [0045], [0078]-[0084], is executed within an isolated container, as described by Jeffries, *see* Ex.1005, 10:54-56, 2:42-49. Thus, LaBine's video conferencing software corresponds to *the multi-user interactive software*. *See* §X.B.6 (limitations [1.4.2], [1.7]); Ex.1003, ¶185.

LaBine further explains that its system allows *a remote machine* to control the host machine. Using LaBine's "Bridgit Conferencing Software," the host computer "treats the remote user input events," such as mouse or keyboard inputs,

“as though they were input locally.” Ex.1006, [0084]. Thus, LaBine’s “client, or ‘participant’ computers” are *remote machine[s]* that have remote control over portions of the host computer. Ex.1003, ¶186.

Through LaBine’s “Bridgit Conferencing Software,” LaBine’s client computer (*a remote machine*) is also allowed to control functions of applications running on the host computer. Ex.1006, [0045]; Ex.1003, ¶187.

For example, LaBine’s remote viewers may access and control the Microsoft Word application of the host computer. *See, e.g.*, Ex.1006, Figs. 20-25. At Figure 22, LaBine shows a “GUI available to a viewer that has remote control of a presenter’s shared desktop.” Ex.1006, [0081]; Ex.1003, ¶¶188-189.

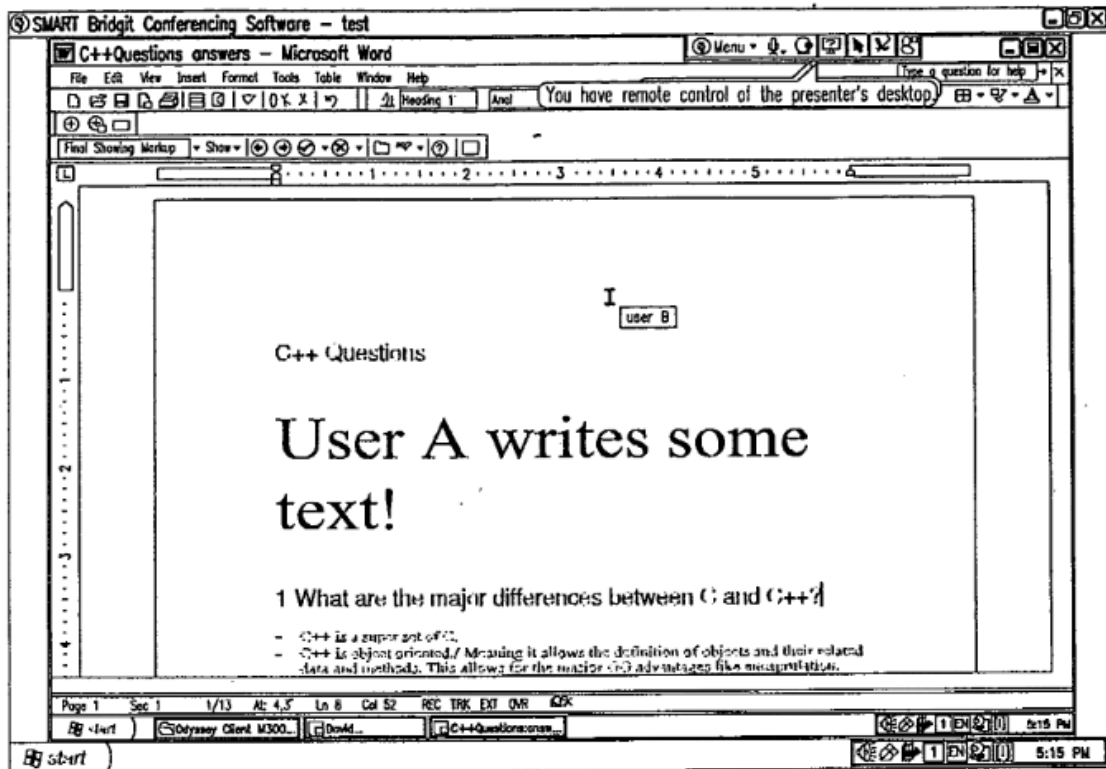


**FIG. 22**

Ex.1006, Fig. 22

In the example shown in Figures 22-25, the remote user remotely modifies a document open in the Microsoft Word application of the host computer. *See* Ex.1006, Figs. 22-25; Ex.1003, ¶189.

In the combination, the Microsoft Word application would correspond to an application running within the isolated containers 122 of Jeffries because it is accessed by untrusted sources. *See* Ex.1005, 10:59-60 (Applications within isolated containers 122 “are thus useable to access untrusted resources.”). In addition, Jeffries’ system enforces isolation between its containers. *See* Ex.1005,

9:67-10:5 (describing preventing one container 122 from accessing another).

However, Jeffries explains that multiple applications may be run in a single container, as shown below. *See* Ex.1005, Fig. 1; Ex.1003, ¶190.

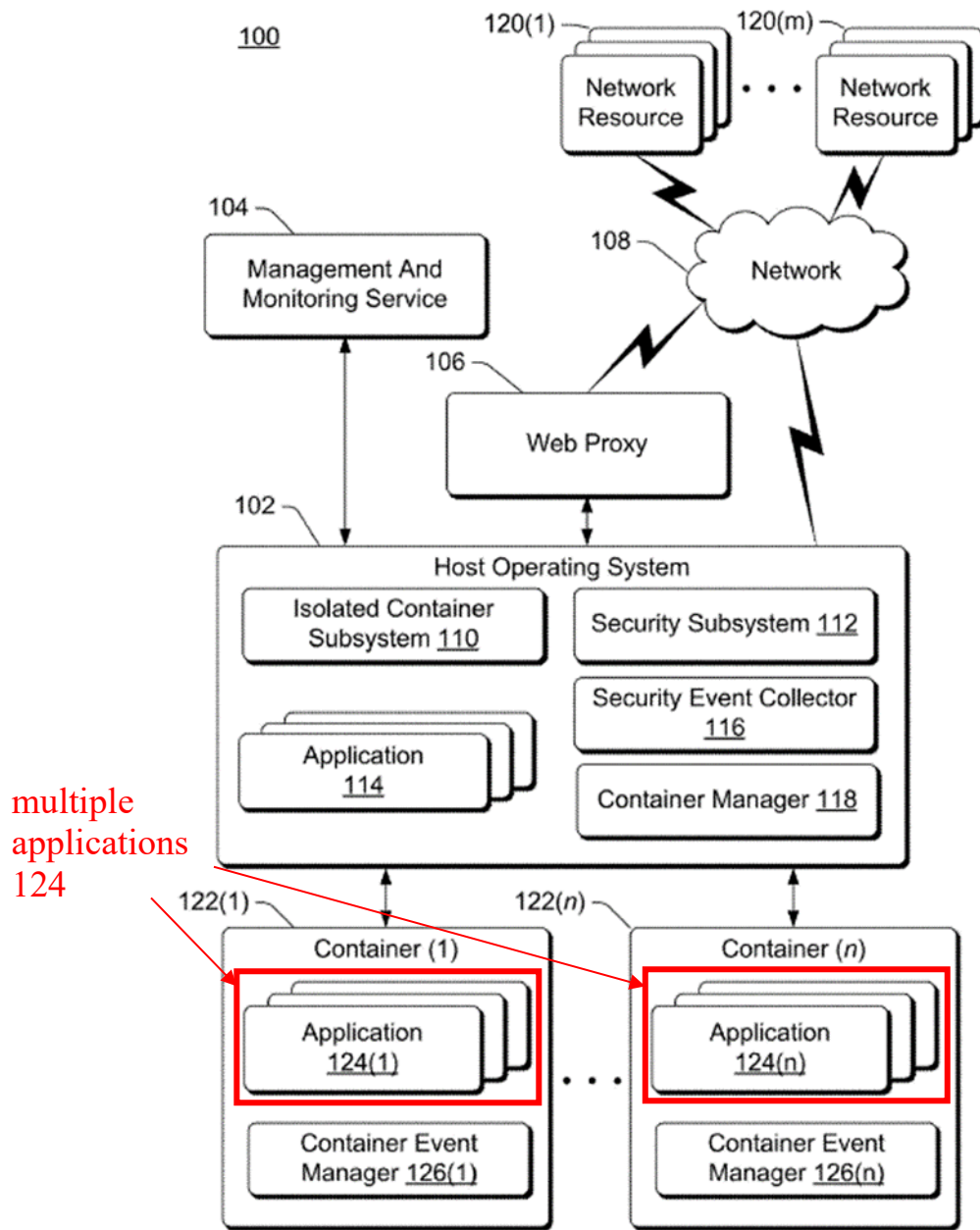


**Fig. 1**

**Ex.1005, Fig. 1 (annotated)**

Accordingly, the Microsoft Word application accessed via LaBine's Bridgit Conferencing Software would be run in the same container as LaBine's Bridgit

Conferencing Software. Ex.1003, ¶191

Thus, LaBine’s Bridgit Conferencing Software (*the multi-user interactive software application*) allows a remote user of a client computer (*allows a remote machine*), to modify a Microsoft Word application (*to control an application or a process of the second set of one or more applications or processes*). In addition, the software would be executed in an isolated container, as described by Jeffries (*running in the isolated computing environment*). Ex.1003, ¶192.

**[2.2] 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 combination of Jeffries and LaBine renders obvious this limitation.  
Ex.1003, ¶193.

LaBine describes *viewing...data from the* applications on the host computer, such as a Microsoft Word application. For example, LaBine’s “GUI 100” allows a remote user to view “the shared region 116 of the presenter’s desktop” which may display an application, such as the Microsoft Word application described above and corresponds to *viewing...data from the application or the process*. Ex.1006, [0049], Fig. 22; Ex.1003, ¶¶194-195.

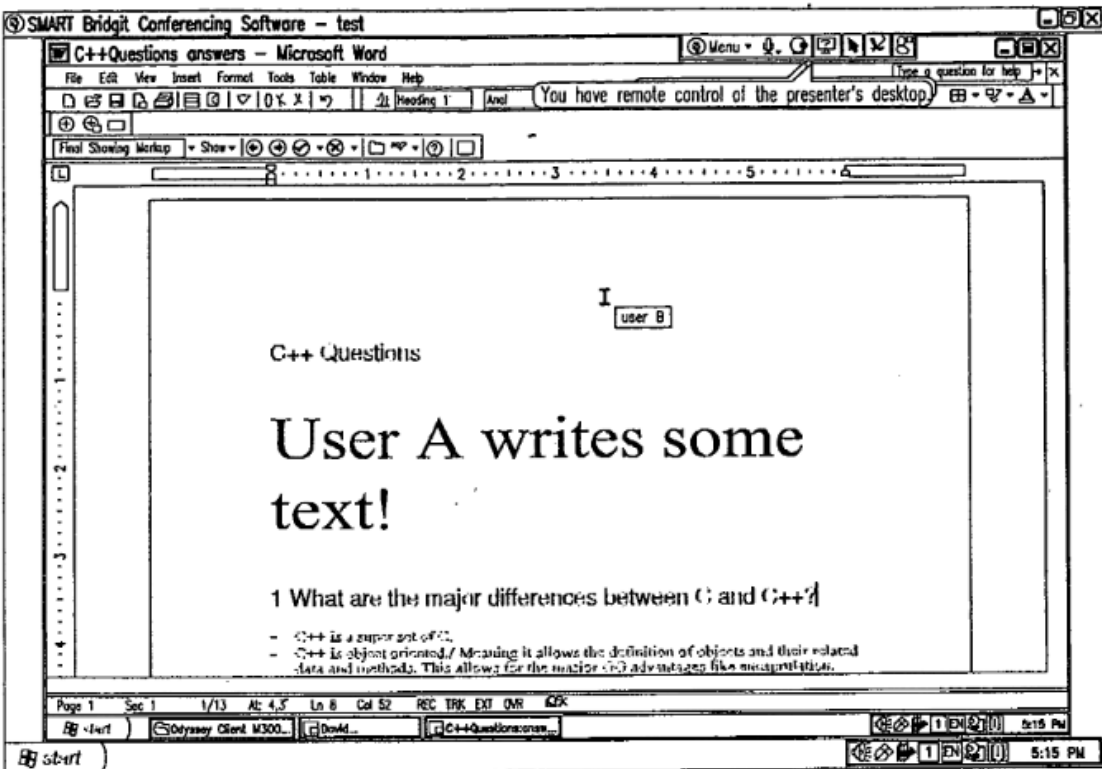


**FIG. 22**

Ex.1006, Fig. 22

In addition, LaBine describes allowing a user of a client computer to input text to a Microsoft Word application which corresponds to *interacting with* and *modifying...data from the application or process*. Ex.1003, ¶196.

Thus, LaBine describes allowing a user of a client computer to view and modify a Microsoft Word application running in a container of the host computer (*wherein controlling the application or process comprises at least one of viewing, interacting with, modifying...data from the application or the process*). Ex.1003, ¶197.

8. **Claim 3**

**[3.0] *The system of claim 1,***

The combination of Jeffries, LaBine, and Ishaya renders obvious *the system of claim 1*. Ex.1003, ¶198.

**[3.1] *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 combination of Jeffries and LaBine renders this limitation obvious. Ex.1003, ¶199.

As described at limitation [2.1] above, the computers of remote users in LaBine are *remote machine[s]*. Because a remote machine is also a remote device, the combination of Jeffries and LaBine render obvious *a remote device* for the same reasons. Ex.1003, ¶200.

In LaBine, “[t]he viewer’s mouse events, including button clicking and movement, are handled as if they occurred locally on the presenters [*sic*] computer.” Ex.1006, [0074]. A mouse is *a pointer device*. LaBine also allows a remote user to control “keyboard...events” which correspond to *keyboard input[s]*. Ex.1006, [0084]. As explained at limitation [2.1], a user of a client computer may also control *a display* by annotating the desktop of the host computer using LaBine’s drawing tool. Ex.1003, ¶201.

Therefore, Jeffries and LaBine describe the Bridgit Conferencing Software

(*the multi-user interactive software application*) allows a viewer (*allows a remote device*) to control mouse events (*to control one or more of a pointer device*), keyboard events (*a keyboard input*), and annotations to the desktop display of the host computer (*or a display of the system*). Ex.1003, ¶202.

## 9. Claim 4

### [4.0] *The system of claim 1,*

The combination of Jeffries, LaBine, and Ishaya renders obvious *the system of claim 1*. Ex.1003, ¶203.

**[4.1] *wherein the system is connected 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.***

Jeffries renders this limitation obvious. Ex.1003, ¶204.

Jeffries explains that “the computing device implementing host operating system 102 accesses network resources 120(1),...,120(m) **via network 108.**”

Ex.1005, 6:59-60. The “[n]etwork 108 represents functionality of a data network, such as the Internet, [or] **a local area network (LAN).**” Ex.1005, 6:57-64; Ex.1003, ¶205.

Jeffries also describes “**a cellular or Wi-Fi network,**” Ex.1005, 25:65-67, and “**a cloud computing environment,**” Ex.1005, 3:24-26; Ex.1003, ¶¶206-207.

Thus, Jeffries’ system may be connected to a LAN network (*wherein the*

*system is connected to a network, the network corresponding to one or more of a local area network (LAN)), a cellular network (a cellular network), a Wi-Fi network (a Wi-Fi network), or a cloud computing environment (cloud computing network).* Ex.1003, ¶2078

#### 10. Claim 7

**[7.0] *The system of claim 4, wherein the processor is further configured to:***

The combination of Jeffries, LaBine, and Ishaya renders obvious *the system of claim 4*, as well as a *processor*. See §X.B.6 [1.2]; Ex.1003, ¶209.

**[7.1] *run one or more trusted applications in the first memory space, and***

Jeffries renders this limitation obvious. Ex.1003, ¶210.

As described at limitation [1.3], Jeffries’ “one of applications 114” correspond to *a first set of one or more applications or processes via a first memory space*. The applications 114 are trusted applications. Jeffries explains that “applications 114...**access a trusted resource** on the host operating system 102.” Ex.1005, 10:59-63. For example, an email client of the host operating system 102 may access a trusted email server. Similarly, a web browser on the host operating system 102 may access a trusted intranet web server. *See, e.g.*, Ex.1005, 6:57-64 (accessing “a local area network (LAN)” or “intranet” “independent of a web proxy”); Ex.1003, ¶211.

Thus, Jeffries describes applications 114 that are not contained in isolated

containers and are used to access trusted resources (*run one or more trusted applications in the first memory space*). It would have been obvious for these applications 114 to be *trusted applications* because if they were not trusted, the user would run them in one of Jeffries' containers. Ex.1003, ¶212.

**[7.2] *wherein the one or more trusted applications communicate with one or more trusted machines via the network.***

Jeffries renders this limitation obvious. Ex.1003, ¶213.

As previously explained at limitations [1.3] and [7.1], Jeffries' applications 114 access trusted resources. *See* §§X.B.6 [1.3], X.B.10 [7.1]. Jeffries also explains that these applications 114 communicate with other *trusted machines via the network*. Ex.1003, ¶214.

At Figure 6, Jeffries provides an additional view of the system 100, in the context of an "example system generally at 600" including a "computing device 602." Ex.1005, 22:39-42. "The platform 624 may...connect the computing device 602 with other computing devices." Ex.1005, 26:1-3. Jeffries further explains that connection with these other computing devices may occur "through a network." Ex.1005, 6:57-64, 25:65-67, 3:24-27, 25:14-18. Thus, Jeffries' system is configured to *communicate with one or more...machines via the network*. Ex.1003, ¶215.

Any of these machines would be *trusted machines*. In Jeffries, these

machines would be part of a local intranet operated by the same company as computed device 602. *See* Ex.1005, 3:19-24 (“multiple users of the enterprise are connected and share access to common enterprise data and/or networks.”). Because these devices are within the local enterprise environment, they are trusted devices. Ex.1003, ¶216.

In addition, Jeffries explains that applications within this enterprise environment access shared, trusted resources. The administrators of a system maintain a list of “resources that are expected to be used by the installed applications.” Ex.1005, 3:30-31. In one example, therefore, an application 114 of the host operating system 102 would be used to access another machine, such as an intranet server of a local network, or a corporate email server, which would be trusted. *See* Ex.1005, 6:64-67; Ex.1003, ¶217.

Thus, Jeffries describes applications 114 executed outside of isolated containers 122 and used to access trusted resources (*the one or more trusted applications*) that communicate with known resources that are expected to be used by the applications 114 in a local enterprise environment (*communicate with one or more trusted machines via the network*). Ex.1003, ¶218.

## **11. Claim 9**

### **[9.0] *A method comprising:***

Jeffries discloses “[a] *method*.” Ex.1003, ¶219.

At Figure 3, Jeffries depicts “**an example process** for implementing isolated container event monitoring.” Ex.1005, 2:23-25; *see also* Ex.1005, Figs. 4-5, 2:26-31, 26:25-30:31; Ex.1003, ¶¶220-221.

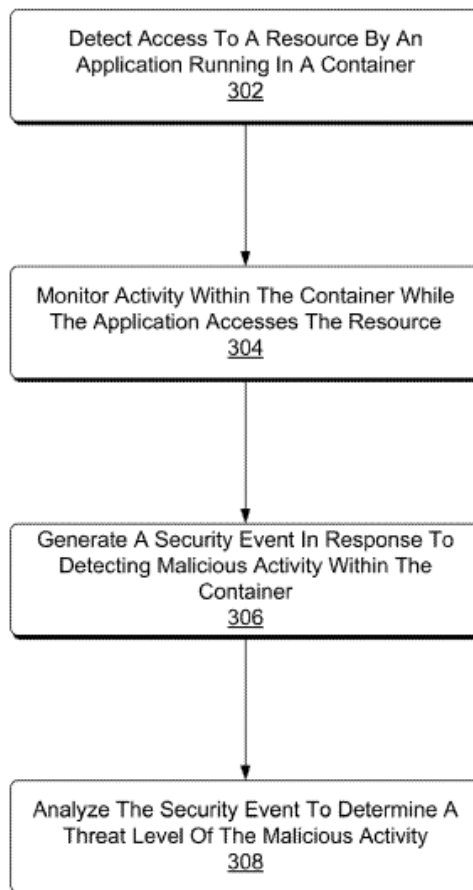


**Fig. 3**

Ex.1005, Fig. 3

Thus, Jeffries discloses “[a] *method*.” Ex.1003, ¶222.

**[9.1] *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;***

Limitation [9.1] is substantially the same as limitation [1.3]. Jeffries renders limitation [9.1] obvious for the same reasons described at limitation [1.3]. *See* §X.B.6, [1.3]; Ex.1003, ¶223.

**[9.2.1] *implementing an isolated computing environment that uses the host operating system, wherein the isolated computing environment enables operation of a second set of one or more applications or processes via a second memory space,***

Limitation [9.2.1] is substantially the same as limitation [1.4.1]. Jeffries renders limitation [9.2.1] obvious for the same reasons described at limitation [1.4.1]. *See* §X.B.6, [1.4.1]; Ex.1003, ¶224.

**[9.2.2] *wherein the second set of one or more applications or processes comprises a multi-user interactive software application;***

Limitation [9.2.2] is the same as limitation [1.4.2] and the combination of Jeffries and LaBine renders limitation [9.2.2] obvious for the same reasons described at limitation [1.4.2]. *See* §X.B.6, [1.4.2]; Ex.1003, ¶225.

**[9.3] *isolating the isolated computing environment from the workspace using an internal isolation firewall;***

Limitation [9.3] is substantially the same as limitation [1.5]. Jeffries renders limitation [9.3] obvious for the same reasons described at limitation [1.5]. *See* §X.B.6, [1.5]; Ex.1003, ¶226.

**[9.4] *authenticating the isolated computing environment with an authentication device; and***

Limitation [9.4] is substantially the same as limitation [1.6]. The

combination of Jeffries and Ishaya renders limitation [9.4] obvious for the same reasons described at limitation [1.6]. *See* §X.B.6, [1.6]; Ex.1003, ¶227.

**[9.5] *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.***

Limitation [9.5] is substantially the same as limitation [1.7]. The combination of Jeffries, LaBine, and Ishaya renders limitation [9.5] obvious for the same reasons described at limitation [1.7]. *See* §X.B.6, [1.7]; Ex.1003, ¶228.

## **12. Claim 10**

**[10.0] *The method of claim 9,***

The combination of Jeffries, LaBine, and Ishaya renders obvious the *method of claim 9*. Ex.1003, ¶229.

**[10.1] *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,***

Limitation [10.1] is the same as limitation [2.1] and the combination of Jeffries and LaBine renders limitation [10.1] obvious for the same reasons described at limitation [2.1]. *See* §X.B.7, [2.1]; Ex.1003, ¶230.

**[10.2] *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.***

Limitation [10.2] is the same as limitation [2.2] and the combination of Jeffries and LaBine renders limitation [10.2] obvious for the same reasons

described at limitation [2.2]. See §X.B.7, [2.2]; Ex.1003, ¶231.

### 13. Claim 11

#### [11.0] *The method of claim 9,*

The combination of Jeffries, LaBine, and Ishaya renders obvious the *method of claim 9*. Ex.1003, ¶232.

#### [11.1] *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 isolated computing environment.*

Limitation [11.1] is similar to limitation [3.1]. Limitation [3.1] recites *a display of the system*. Limitation [11.1] recites *a display of the isolated computing environment*. The combination of Jeffries and LaBine renders limitation [11.1] obvious for the same reasons described at limitation [3.1] as well as those explained below. See §X.B.8, [3.1]; Ex.1003, ¶233.

In the '601 patent, the mouse, keyboard, and display of *the system* is the same as the mouse, keyboard, and display of *the isolated computing environment*. “The host computer” of the '601 patent “include[s] [the] isolated computing environment,” described as a “virtual machine.” Ex.1001, 1:64-65, 5:41-43. As a virtual machine, the isolated computing environment does not use separate hardware (mouse, keyboard, and display) from the host computer. Ex.1003, ¶234.

Similarly, Jeffries’ isolated containers 122 act as virtual machines that use the same hardware as the host computer. See Ex.1005, 10:38-43; §X.B.6, [1.3]-

[1.4.1]; Ex.1003, ¶¶235-236.

Therefore, LaBine's disclosure of allowing inputs from a viewer to be transmitted to the host machine and treated as inputs of the host machine renders obvious *allow[ing] a remote device to control one or more of a pointer device, a keyboard input, or a display of the isolated computing environment*. Ex.1003, ¶237.

#### 14. Claim 12

**[12.0] *The method of claim 9, further comprising***

The combination of Jeffries, LaBine, and Ishaya renders obvious the *method of claim 9*. Ex.1003, ¶238.

**[12.1] *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.***

Limitation [12.1] is substantially the same as limitation [4.1]. Jeffries renders limitation [12.1] obvious for the same reasons described at limitation [4.1]. See §X.B.9, [4.1]; Ex.1003, ¶239.

#### 15. Claim 15

**[15.0] *The method of claim 12, further comprising:***

The combination of Jeffries, LaBine, and Ishaya renders obvious the *method of claim 12*. Ex.1003, ¶240.

**[15.1] *running one or more trusted applications in the first memory space,***

Limitation [15.1] is substantially the same as limitation [7.1]. Jeffries renders limitation [15.1] obvious for the same reasons described at limitation [7.1]. See §X.B.10, [7.1]; Ex.1003, ¶241.

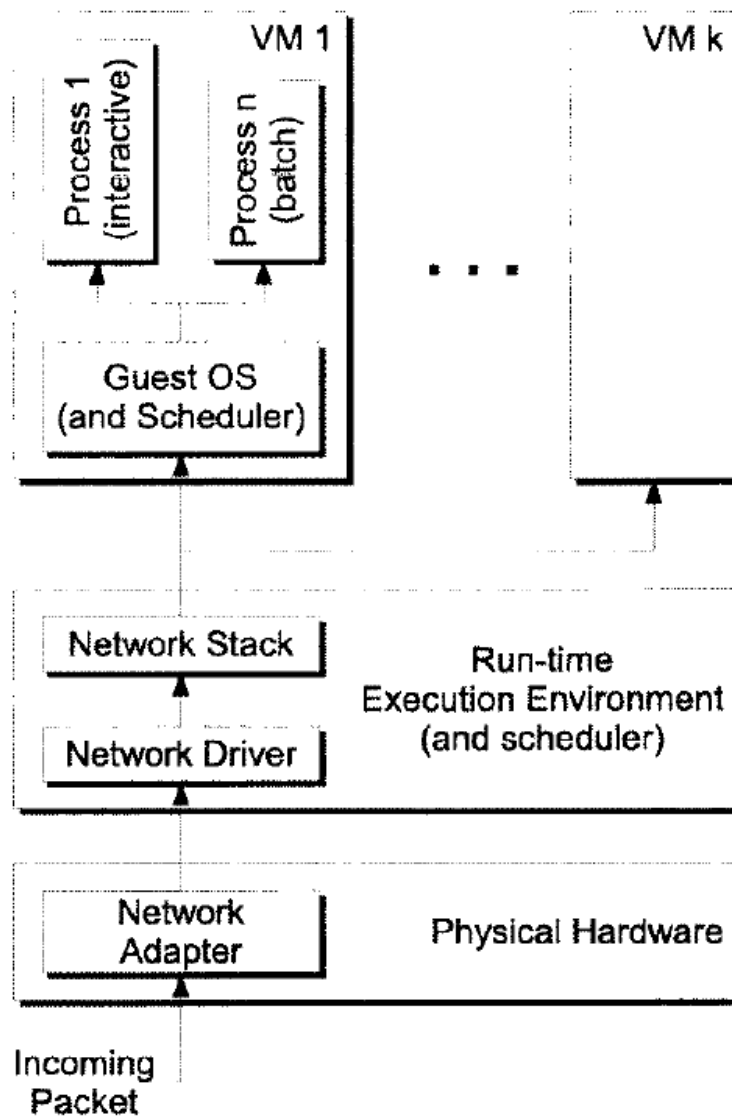
**[15.2] *wherein the one or more trusted applications communicate with one or more trusted machines via the network.***

Limitation [15.2] is the same as limitation [7.2] and Jeffries renders limitation [15.2] obvious for the same reasons described at limitation [7.2]. See §X.B.10, [7.2]; Ex.1003, ¶242.

**C. Ground 2: Claims 5-6 and 13-14 are obvious in view of Jeffries, LaBine, Ishaya, and Cucinotta.**

**1. Summary of Cucinotta.**

Cucinotta describes a computer system that implements isolated computing environments, referred to as “virtual machines.” Ex.1008, [0002]. Cucinotta’s system is illustrated in Figure 1. Ex.1003, ¶243.



Ex.1008, Fig. 1

Cucinotta's system includes a "firewall component." Ex.1008, [0039]. The firewall component "inspects each and every incoming and outgoing packet" for the host system as well as "traffic...directed towards KVM handling VMs." Ex.1008, [0039], [0049]; Ex.1003, ¶244.

Because Cucinotta implements multiple isolated computing environments, it

is in the same field of endeavor and is reasonably pertinent to a particular problem with which the inventor of the '601 patent was involved. Ex.1003, ¶245.

**2. Reasons to Combine Cucinotta with Jeffries, LaBine, and Ishaya.**

A POSITA would have combined Jeffries, LaBine, and Ishaya with Cucinotta because both the combination described above and Cucinotta describe improving system security and isolating applications. Jeffries isolates applications that access untrusted resources in a container. *See* Ex.1005, 2:46-49, 10:59-60. Cucinotta runs “processes” in virtual machines. Ex.1008, Fig. 1 (illustrating VM 1-VM k), [0002]; Ex.1003, ¶246.

A POSITA would have been motivated to implement a “firewall component” described in Cucinotta to further protect the system. Ex.1008, [0039]. As explained above, isolating applications that access untrusted resources in containers prevents the effects of malware from spreading outside the container. *See* Ex.1005, 2:46-49, 10:59-60. However, this method protects applications from potential attacks directed to the isolated containers. A POSITA would be motivated to additionally implement a firewall component to the workspace to protect applications and resources not protected by isolated containers. Ex.1003, ¶247.

In addition, a POSITA would be motivated to implement a host-based firewall, as described in Cucinotta, to prevent malicious files from being received

within isolated containers in the first place. Jeffries explains the advantages of isolating applications. *See* Ex.1005, 2:42-49, 10:59-63. However, protection through an isolated container typically requires terminating the container, along with its contents, to prevent the spread of malware. *See* Ex.1005, 16:37-42 (“terminating the container 122 and rebuilding a new container”), 27:19-41. Doing so results in the loss of unsaved work and is a lengthy process leading to frustration. Implementing a host-based firewall would minimize the instances in which an isolated container must be terminated and rebuilt by preventing malicious files from infecting an isolated container. Ex.1003, ¶248.

The combination of Jeffries, LaBine, and Ishaya with Cucinotta would be simply combining prior art elements according to known methods to yield predictable results. The known methods would involve implementing a host-based firewall as described in Cucinotta, *see* Ex.1008, [0039], [0049], within a system that utilizes isolated containers as described in both Jeffries and Cucinotta, *see* Ex.1005, Abstract, 2:42-49; Ex.1008, Fig. 1. The results of the combination would have been predictable because Cucinotta’s system includes the host-based firewall component and has a similar system architecture as Jeffries. *See* Ex.1005, Fig. 1; Ex.1008, Fig. 1; Ex.1003, ¶249.

Benefits of the combination include improving the security of a computer system by protecting all components of a host system, both within and outside of

isolated environments. In addition, implementing a host-based filter would enhance efficiency by filtering packets on behalf of the entire host system. A host-based firewall may also advantageously maintain consistency and uniformity of packet filtering policies across multiple isolated environments. Ex.1003, ¶250.

A POSITA would have had a reasonable expectation of success in implementing the firewall described by Cucinotta in the combined system. The host-based firewall component in Cucinotta inspects and filters packets destined for the virtual machines 1-k and host system. Ex.1008, [0049]. Such a host-based filter would be readily implemented to similarly inspect and filter packets destined for the containers 122 and host system 102 in Jeffries. *See* Ex.1005, Fig. 1; Ex.1003, ¶251.

Accordingly, a POSITA would be motivated to implement Cucinotta's host-based firewall component with the combined system. The combination relies on the combined teachings of the references as understood by a POSITA and not on physical incorporation of elements. *In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012); Ex.1003, ¶252.

### **3. Claim 5**

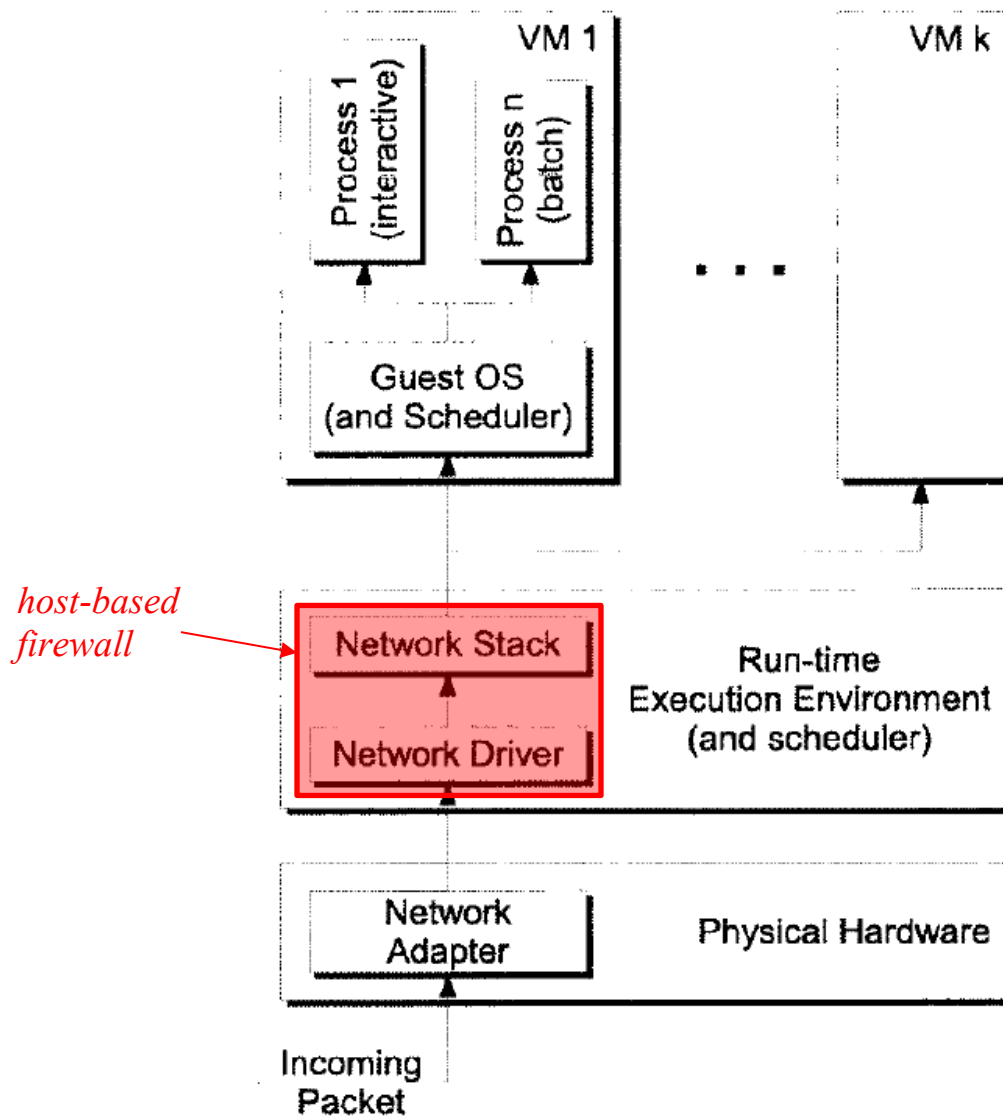
**[5.0] *The system of claim 4, wherein the processor is further configured to:***

The combination of Jeffries, LaBine, and Ishaya renders obvious the *system of claim 4*. Ex.1003, ¶253.

**[5.1] *implement a host-based firewall, wherein the host-based firewall isolates the system from communicating with one or more other devices on the network.***

The combination of Jeffries, LaBine, Ishaya, and Cucinotta renders this limitation obvious. Ex.1003, ¶254.

Cucinotta describes “a software component” that “comprise[s] a firewall component.” Ex.1008, [0039]. This firewall is part of “a multi-programmed computing system.” Ex.1008, [0001]. Figure 1, below, illustrates the path of a network packet received by the system. Ex.1008, [0032]. Cucinotta’s “firewall component” may “belong to the network stack of the runtime, or may be a network driver,” highlighted below. Ex.1008, [0039]; Ex.1003, ¶255.



Ex.1008, Fig. 1 (annotated)

Because Cucinotta’s software component resides within the computing system shown in Figure 1, it is a *host-based firewall*. In addition, Cucinotta’s software component isolates the entire computer system from communicating with other devices because it “inspect[s] each and every incoming and outgoing packet” Ex.1008, [0039]. Cucinotta’ “firewall software is typically operable to intercept all

incoming and outgoing traffic travelling across [the] computing system.” Ex.1008, [0049]; Ex.1003, ¶256.

Intercepting traffic isolates the host computer from external resources. Filters at the time commonly used whitelist and blacklist features to isolate protected machines or regions of memory. *See, e.g.*, Ex.1023, [0143] (“blacklisting...malicious sites”); Ex.1029, Abstract (“creat[ing] and manag[ing] whitelists 111 and blacklists 116 which are used as filters”). The ’601 patent describes isolating systems using similar whitelist and blacklist features. *See* Ex.1001, 6:28-34 (describing “a list of untrusted network destinations” as “a blacklist”); Ex.1003, ¶257.

In the combination, a host-based firewall component would be implemented with Jeffries’ host operating system 102. As described in Cucinotta, this host-based firewall component would inspect all packets received by the host operating system 102 and selectively allow packets based on security policies. *See* Ex.1008, [0049]; Ex.1003, ¶258.

Thus, the combination describes implementing a firewall component (*implement a host-based firewall*) that isolates the host computer system by intercepting all incoming and outgoing packets (*wherein the host-based firewall isolates the system from communicating with one or more other devices on the network*). Ex.1003, ¶259.

4. **Claim 6**

**[6.0] *The system of claim 5,***

The combination of Jeffries, LaBine, Ishaya, and Cucinotta renders obvious the *system of claim 5*. Ex.1003, ¶260.

**[6.1] *wherein the host-based firewall blocks incoming communications from the network that are sent to the workspace.***

The combination of Jeffries, LaBine, Ishaya, and Cucinotta renders this limitation obvious. Ex.1003, ¶261.

Jeffries describes a *workspace*. See §X.B.6, [1.3]. In the combination, a host-based filter would be implemented in conjunction with the host computer of Jeffries to protect the workspace. See §X.C.3, [5.1]. Packet filters block incoming communications. See Ex.1027, [0015] (“packet filter firewall could block incoming packets”); Ex.1028, 6:39-45 (“a packet or circuit filter...blocks all incoming...packets except for well-known Domain Name Service (‘DNS’) packets”). Cucinotta’s firewall component is configured to “intercept all incoming and outgoing traffic travelling across [the] computing system.” Ex.1008, [0049]. In the combination, a host-based packet filter would selectively block packets to or from the host computer. The firewall component filters all packets sent either to the workspace or the isolated containers 122 of Jeffries. Ex.1008, [0049] (“intercept any incoming...traffic...including instances when such traffic is

directed towards...VMs”); Ex.1003, ¶262.

Thus, the combination describes a firewall component (*the host-based firewall*) that intercepts (*blocks*) network packets from external devices (*incoming communications from the network*) from reaching the workspace (*that are sent to the workspace*). Ex.1003, ¶263.

## 5. Claim 13

**[13.0] *The method of claim 12, further comprising,***

The combination of Jeffries, LaBine, Ishaya, and Cucinotta renders obvious the *method of claim 12*. Ex.1003, ¶264.

**[13.1] *implementing a host-based firewall, wherein the host-based firewall isolates the first memory space and the second memory space from communicating with one or more devices on the network.***

Limitation [13.1] is substantially similar to limitation [5.1]. Limitation [13.1] recites *the host-based firewall isolates the first memory space and the second memory space*, whereas limitation [5.1] recites *the host-based firewall isolates the system*. Ex.1003, ¶265.

As explained at limitation [1.3], Jeffries describes “memory locations” associated with applications 114 (*the first memory space*) and separate memory associated with applications 124 on isolated containers 122 (*the second memory space*). See Ex.1005, 12:11-16, 15:26-30; §§X.B.6, [1.3], [1.4.1]; Ex.1003, ¶266.

Because the “memory locations” (*the first memory space*) and the “memory”

allocated to the isolated containers 122 (*the second memory space*) are part of the same host computer in Jeffries, the host-based firewall described at [5.1] isolates both *the first memory space* and *the second memory space*. But additionally, Cucinotta describes intercepting packets addressed to the VMs of its system. *See* Ex.1008, [0049] (“firewall software...can also be used in combination with a KVM hypervisor”). In such case, “iptables operates to intercept any incoming, outgoing, or forwarded traffic travelling across a Linux system, including instances when such traffic is **directed towards KVM handling VMs.**” Ex.1008, [0049]; Ex.1003, ¶267.

In the combination, the firewall component described by Cucinotta and implemented within Jeffries’ system would similarly intercept packets destined for the host computer, corresponding to *the first memory space*, as well as the isolated containers 122, corresponding to *the second memory space*. Ex.1003, ¶268.

Thus, the combination renders obvious implementing a firewall component (*implementing a host-based firewall,*), that intercepts packets destined for the host computer (*wherein the host-based firewall isolates the first memory space*) as well as packets destined for the isolated containers 122 (*and the second memory space*) to prevent communication with external devices (*from communicating with one or more devices on the network*). Ex.1003, ¶269.

**6. Claim 14**

**[14.0] *The method of claim 13,***

The combination of Jeffries, LaBine, Ishaya, and Cucinotta renders obvious the *method of claim 13*. Ex.1003, ¶270.

**[14.1] *wherein the host-based firewall blocks incoming communications from the network that are sent to the workspace.***

Limitation [14.1] is the same as limitation [6.1] and the combination of Jeffries and Cucinotta renders limitation [14.1] obvious for the same reasons described at limitation [6.1]. *See* §X.C.4, [6.1]; Ex.1003, ¶271.

**D. Ground 3: The combination of Jeffries, LaBine, Ishaya, and Levy renders obvious claims 8 and 16.**

**1. Summary of Levy.**

Levy describes a “browser operating system (BOS)” that “runs...each Web application (e.g., on-line banking, and Web mail) in its own virtual machine, which provides strong isolation between Web services and the user’s local resources.”

Ex.1009, Abstract. This isolation “protect[s] other applications and client resources from malicious downloaded code.” Ex.1009, [0057], [0130], [0046], [0072]; Ex.1003, ¶272.

Levy additionally “empower[s]...users” by providing “explicit knowledge about and control over downloaded content and code.” Ex.1009, [0057], Abstract. Each virtual machine has “a private temporary storage.” Ex.1009, [0020]. “A user

is enabled to explicitly initiate a transfer of an object into or out of the temporary storage.” Ex.1009, [0020]. Levy is analogous art to the ’601 patent. Like the ’601 patent, Levy executes applications in isolated environments to prevent malware received from infecting the rest of the system. *See* Ex.1009, [0057], [0130], [0046], [0072]. Thus, Levy is in the same field of endeavor and is reasonably pertinent to a particular problem with which the inventor of the ’601 patent was involved (i.e., system security). Ex.1003, ¶¶273-274.

## **2. Reasons to Combine Levy with Jeffries, LaBine, and Ishaya.**

A POSITA would have combined Jeffries, LaBine, and Ishaya with Levy because Levy similarly describes isolating applications that access untrusted resources. Both Jeffries and Levy describe isolated environments. *See* Ex.1005, 2:42-49, 10:59-63 (containers); Ex.1009, [0014], [0021], [0039], [0043] (virtual machines); Ex.1003, ¶275.

A POSITA would have been motivated to allow the user to designate which files may be transferred from an isolated environment to the host computer, as taught by Levy. Levy’s system “provides strong safety guarantees both to users and Web services” while “enabl[ing] users to gain knowledge and control of the active Web content being downloaded and executed.” Ex.1009, [0014], [0052]. A POSITA would be motivated to implement this technique in the combined system described above. For example, while running a video conferencing software in an

isolated container, the user of the host computer may receive a link from a client computer through the video conferencing software. Because this link is untrusted, it may be opened in an “instance[]” of a web browser within the isolated container. Ex.1005, 10:57; *see supra* §X.B.7 (limitation [2.1]) (describing multiple applications in a single isolated container). If, however, the user determines that the content received is trustworthy, the user may designate that it be transferred out of the isolated container to the remainder of the system, as taught by Levy. Ex.1009, [0020]; Ex.1003, ¶276.

Implementing Levy’s teachings in the combined system would advantageously reduce false positives of files incorrectly flagged as malicious. In some scenarios, automatic filters may flag a trusted item as untrusted. For example, if the item were sent by the same user from a public machine, the host computer may flag the item as untrusted and isolate it. It would be advantageous to allow the user to designate objects to be released from the isolated container. Ex.1003, ¶277.

The combination would be simply combining prior art elements according to known methods to yield predictable results. The known methods including designating data objects to be transferred from the container. The results would be predictable because Levy implements this feature in a substantially similar system. Ex.1009, [0020]; Ex.1003, ¶278.

The combination would improve methods of conveying security state

information to a user. The combination would also optimize resource management, ensuring that only necessary files are transferred from the isolated container, preserving memory availability. This process also raises security awareness of users and adds greater flexibility and customization. For example, administrators or policies may inspect files to be transferred. Ex.1003, ¶¶279-280.

A POSITA would have had a reasonable expectation of success in implementing Levy's techniques with the combined system. Levy's system is designed to work with "conventional window systems." Ex.1009, [0052]. Levy's "BOS is designed to be implement[ed] in different ways." Ex.1009, [0056]. Therefore, a POSITA would have recognized that Levy's teachings would be readily incorporated with the combined system. Ex.1003, ¶281.

For all these reasons, a POSITA would be motivated to implement Levy's teachings in the combined system. Doing so would predictably improve the security of the host computer by providing the user with additional information as to the source of downloaded files, as well as reduce false positives that may arise from automated or absolute isolation. The combination relies on the combined teachings of the references as understood by a POSITA and not on a physical incorporation of elements from one into another. *In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012); Ex.1003, ¶282.

3. **Claim 8.**

**[8.0] *The system of claim 1,***

The combination of Jeffries, LaBine, and Ishaya renders obvious the *system of claim 1*. Ex.1003, ¶283.

**[8.1] *wherein the internal isolation firewall prompts a user of the system to allow communication between the first memory space and the second memory space.***

The combination of Jeffries, LaBine, Ishaya, and Levy renders this limitation obvious. Ex.1003, ¶284.

Levy's system uses virtual machines that are remarkably similar to both the '601 patent and Jeffries. *See* Ex.1009, Abstract, [0057], [0130]; §X.D.1-2; Ex.1003, ¶285.

In Levy, each virtual machine has a “temporary storage” or “holding bin.” Ex.1009, [0020], [0067]. “This private temporary storage is used for temporarily holding an object that is to be transferred into or out of the virtual machine.” Ex.1009, [0020]. “A user is enabled to explicitly initiate a transfer of an object into or out of the temporary storage.” Ex.1009, [0020]; *see also* Ex.1009, [0067]; Ex.1003, ¶¶286-287.

In the combination, a POSITA would have been motivated to similarly allow a user to control the transfer of objects, as taught by Levy, between the memory associated with the isolated containers 122 and the remaining, unrestricted memory

of the host operating system 102 in Jeffries. This functionality would be implemented within container manager 118 described in Jeffries, which implements a “policy describing what types of access are authorized or unauthorized.” Ex.1005, 17:35-40; Ex.1003, ¶288.

Thus, the combination of Jeffries and Levy describes the container manager 118 (*internal isolation firewall*) allowing a user to explicitly initiate a transfer of the data (*prompts a user of the system to allow communication*) from memory associated with the isolated containers 122 to the memory of the remainder of the system (*between the first memory space and the second memory space*). Ex.1003, ¶289.

#### 4. Claim 16.

##### [16.0] *The system of claim 9,*

The combination of Jeffries, LaBine, and Ishaya renders obvious the *method of claim 9*. Ex.1003, ¶290.

##### [16.1] *wherein the internal isolation firewall prompts a user of the workspace to allow communication between the first memory space and the second memory space.*

Limitation [16.1] is substantially similar to limitation [8.1]. While limitation [16.1] recites *wherein the internal isolation firewall prompts a user of the workspace*, limitation [8.1] recites *prompts a user of the system*. The combination of Jeffries and Levy renders this limitation obvious for the same reasons as

described at limitation [8.1]. *See* § X.D.3, [8.1]; Ex.1003, ¶291.

In addition, the combination of Jeffries and Levy renders obvious *prompt[ing] a user of the **workspace*** because the claimed *system* comprises the *workspace*. *See* §X.B.6, [1.3]. Therefore, a *user of the system* is also the *user of the workspace*. Ex.1003, ¶292.

## **XI. CONCLUSION**

Accordingly, Petitioner has established a reasonable likelihood that the Challenged Claims are unpatentable.

Respectfully submitted,

Dated: August 13, 2024  
HAYNES AND BOONE, LLP  
2801 N. Harwood St., Suite 2300  
Dallas, Texas 75201  
Customer No. 27683

/Theodore M. Foster/  
Theodore M. Foster  
Lead Counsel for Petitioner  
Registration No. 57,456

## **XII. MANDATORY NOTICES**

### **A. Real Party-in-Interest**

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner certifies that the real party-in-interest is Cisco Systems, Inc.

### **B. Related Matters**

Pursuant to 37 C.F.R. § 42.8(b)(2), to the best knowledge of the Petitioner, the '601 patent is involved in the following case:

| <b>Case Heading</b>                                                           | <b>Number</b> | <b>Court</b> | <b>Date</b>       |
|-------------------------------------------------------------------------------|---------------|--------------|-------------------|
| <i>Croga Innovations Ltd. v. Cisco Systems, Inc.</i>                          | 2-24-cv-65    | EDTX         | February 1, 2024  |
| <i>Croga Innovations Ltd. v. International Business Machines Corporation.</i> | 2-23-cv-00634 | EDTX         | December 29, 2023 |
| <i>Croga Innovations Ltd. v. Fortinet, Inc.</i>                               | 2-24-cv-00206 | EDTX         | March 22, 2024    |
| <i>Croga Innovations Ltd. v. Amazon Web Services, Inc.</i>                    | 1-24-cv-00398 | WDTX         | April 16, 2024    |

### **C. Lead and Back-up Counsel and Service Information**

#### Lead Counsel

Theodore M. Foster  
HAYNES AND BOONE, LLP  
2801 N. Harwood St., Suite 2300  
Dallas, TX 75201

Phone: (303) 382-6205  
Fax: (214) 200-0853  
ipr.theo.foster@haynesboone.com  
USPTO Reg. No. 57,456

Back-up Counsel

David L. McCombs  
HAYNES AND BOONE, LLP  
2801 N. Harwood St., Suite 2300  
Dallas, TX 75201

Phone: (214) 651-5533  
Fax: (214) 200-0853  
david.mccombs.ipr@haynesboone.com  
USPTO Reg. No. 32,271

Eugene Goryunov  
HAYNES AND BOONE, LLP  
2801 N. Harwood St., Suite 2300  
Dallas, TX 75201

Phone: (312) 216-1630  
Fax: (214) 200-0853  
eugene.goryunov.ipr@haynesboone.com  
USPTO Reg. No. 61,579

Adam Erickson  
HAYNES AND BOONE, LLP  
2801 N. Harwood St., Suite 2300  
Dallas, TX 75201

Phone: (202) 654-4531  
Fax: (214) 200-0853  
adam.erickson.ipr@haynesboone.com  
USPTO Reg. No. 77,434

Please address all correspondence to lead and back-up counsel. Petitioner consents to service in this proceeding by email at the addresses above.

**CERTIFICATE OF WORD COUNT**

Pursuant to 37 C.F.R. § 42.24(d), Petitioner hereby certifies, in accordance with and in reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,985.

Pursuant to 37 C.F.R. § 42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under § 42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: August 13, 2024

/Theodore M. Foster/

Theodore M. Foster  
Lead Counsel for Petitioner  
Registration No. 57,456

**CERTIFICATE OF SERVICE**

The undersigned certifies that, in accordance with 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105, service was made on Patent Owner as detailed below.

*Date of service* August 13, 2024

*Manner of service* FEDERAL EXPRESS

*Documents served* Petition for *Inter Partes* Review Under 35 U.S.C. § 312 and 37 C.F.R. § 42.104 of U.S. 11,223,601; Petitioner's Exhibit List; All Exhibits; Petitioner's Power of Attorney.

*Persons served* Atlantic IP  
c/o Lombard Geliebter  
1325 Avenue of the Americas, 28<sup>th</sup> Floor  
New York, NY 10019

/Theodore M. Foster/  
Theodore M. Foster  
Lead Counsel for Petitioner  
Registration No. 57,456