PTO/SB/57 (11-23)
Approved for use through 11/30/2024. OMB 0651-0064
U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

(Also referred to as FORM PTO-1465)

REQUEST FOR *EX PARTE* REEXAMINATION TRANSMITTAL FORM

Address to:
Mail Stop *Ex Parte* Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attorney Docket No.: VIR058

Date: September 23, 2024

1. ☒ This is a request for *ex parte* reexamination pursuant to 37 CFR 1.510 of patent number 7,784,058 issued August 24, 2010. The request is made by:
   ☐ patent owner. ☒ third party requester.

2. ☒ The name and address of the person requesting reexamination is:
   Michelle Aspen
   Unified Patents, LLC, 4445 Willard Ave., Suite 600
   Chevy Chase, MD 20815

3. Requester ☐ asserts small entity status (37 CFR 1.27) or ☐ certifies micro entity status (37 CFR 1.29). Only a patent owner requester can certify micro entity status. Form PTO/SB/15A or B must be attached to certify micro entity status.

4. This request is accompanied by payment of the reexamination fee as set forth in:
   ☒ 37 CFR 1.20(c)(2), **or**
   ☐ 37 CFR 1.20(c)(1). **In checking this box for payment of the fee set forth in 37 CFR 1.20(c)(1), requester asserts that this request has forty (40) or fewer pages and complies with all other requirements of 37 CFR 1.20(c)(1).**

   Payment of the reexamination fee is made by the method set forth below:
   a. ☐ A check in the amount of $_____ is enclosed to cover the reexamination fee;
   b. ☐ The Director is hereby authorized to charge the reexamination fee to Deposit Account No. _____;
   c. ☐ Payment by credit card. Form PTO-2038 is attached, **or**
   d. ☒ Payment made via USPTO patent electronic filing system.

   ☒ In addition, the Director is hereby authorized to charge any fee deficiencies to Deposit Account No. 50-6990.

5. ☒ Any refund should be made by ☐ check or ☒ credit to Deposit Account No. 50-6990. 37 CFR 1.26(c). If payment is made by credit card, refund must be to credit card account.

6. ☒ A copy of the patent to be reexamined having a double column format on one side of a separate paper is enclosed. 37 CFR 1.510(b)(4).

7. ☐ CD-ROM or CD-R in duplicate, Computer Program (Appendix) or large table
   ☐ Landscape Table on CD

8. ☐ Nucleotide and/or Amino Acid Sequence Submission
If applicable, items a. -- c. are required.
   a. ☐ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. ☐ CD-ROM (2 copies) or CD-R (2 copies) **or**
      ii. ☐ paper
   c. ☐ Statements verifying identity of above copies.

9. ☒ A copy of any disclaimer, certificate of correction or reexamination certificate issued in the patent is included.

10. ☒ Reexamination of claim(s) 1 and 12 is requested.

11. ☒ A copy of every patent or printed publication relied upon is submitted herewith including a listing thereof on Form PTO/SB/08, PTO-1449, or equivalent.

12. ☐ An English language translation of all necessary and pertinent non-English language patents and/or printed publications is attached.

13. ☒ The attached detailed request includes at least the following items:
   a. A statement identifying each substantial new question of patentability based on prior patents and printed publications. 37 CFR 1.510(b)(1).
   b. An identification of every claim for which reexamination is requested, and a detailed explanation of the pertinency and manner of applying the cited art to every claim for which reexamination is requested 37 CFR1.510(b)(2).

14 ☐ A proposed amendment is included (only where the patent owner is the requester). 37 CFR 1.510(e).

15. ☒ It is certified that the statutory estoppel provisions of 35 U.S.C. 315(e)(1) or 35 U.S.C. 325(e)(1) do not prohibit requester from filing this *ex parte* reexamination request. 37 CFR 1.510(b)(6).

16. Service
   a. ☒ It is certified that a copy of this request (if filed by other than the patent owner) has been served in its entirety on the patent owner as provided in 37 CFR 1.33(c).

      The name and address of the party served are:

      27975 - ADD+G - 27975

      1135 East State Road 434, Suite 3001

      Winter Springs, FL 32708

      Date of Service: September 23, 2024

   OR

   b. ☐ A duplicate copy is enclosed since service on patent owner was not possible. An explanation of the efforts made to serve patent owner **is attached.** See MPEP 2220.

17. Correspondence Address: Direct all communication about the reexamination to:

☒ The address associated with Customer Number: 165774

OR

☐ Firm or Individual Name _____
(at the address identified below)

Address
Unified Patents, LLC, 4445 Willard Avenue, Suite 600

City	State	Zip
Chevy Chase	MD	20815

Country
United States

Telephone	Email
(559) 214-3388	michelle@unifiedpatents.com

18. ☒ The patent is currently the subject of the following concurrent proceeding(s):

a. ☐ Copending reissue Application No. _____

b. ☐ Copending reexamination Control No. _____

c. ☐ Copending Interference No. _____

d. ☒ Copending litigation styled:
See attached list for co-pending cases

WARNING : Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

/Michelle Aspen/ — Authorized Signature
September 23, 2024 — Date

Michelle Aspen — Typed/Printed Name
75,665 — Registration No.

☐ For Patent Owner Requester ☒ For Third Party Requester

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. The United States Patent and Trademark Office (USPTO) collects the information in this record under authority of 35 U.S.C. 2. The USPTO's system of records is used to manage all applicant and owner information including name, citizenship, residence, post office address, and other information with respect to inventors and their legal representatives pertaining to the applicant's/owner's activities in connection with the invention for which a patent is sought or has been granted. The applicable Privacy Act System of Records Notice for the information collected in this form is COMMERCE/PAT-TM-7 Patent Application Files, available in the Federal Register at 78 FR 19243 (March 29, 2013). https://www.govinfo.gov/content/pkg/FR-2013-03-29/pdf/2013-07341.pdf

Routine uses of the information in this record may include disclosure to:

1) law enforcement, in the event that the system of records indicates a violation or potential violation of law;

2) a federal, state, local, or international agency, in response to its request;

3) a contractor of the USPTO having need for the information in order to perform a contract;

4) the Department of Justice for determination of whether the Freedom of Information Act (FOIA) requires disclosure of the record;

5) a Member of Congress submitting a request involving an individual to whom the record pertains, when the individual has requested the Member's assistance with respect to the subject matter of the record;

6) a court, magistrate, or administrative tribunal, in the course of presenting evidence, including disclosures to opposing counsel in the course of settlement negotiations;

7) the Administrator, General Services Administration (GSA), or their designee, during an inspection of records conducted by GSA under authority of 44 U.S.C. 2904 and 2906, in accordance with the GSA regulations and any other relevant (i.e., GSA or Commerce) directive, where such disclosure shall not be used to make determinations about individuals;

8) another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c));

9) the Office of Personnel Management (OPM) for personnel research purposes; and

10) the Office of Management and Budget (OMB) for legislative coordination and clearance.

If you do not furnish the information requested on this form, the USPTO may not be able to process and/or examine your submission, which may result in termination of proceedings, abandonment of the application, and/or expiration of the patent.

- *VirtaMove, Corp. v. Hewlett Packard Enterprise Company*, Case No. 2:24-cv-00093 (E.D. Tex., Feb. 9, 2024)
- *VirtaMove, Corp. v. Amazon.com, Inc. et al*, Case No. 7:24-cv-00030 (W.D. Tex., Jan. 26, 2024)
- *RedHat, Inc. v. VirtaMove, Corp.*, Case No. 5:24-cv-04740 (N.D. Cal., Aug. 5, 2024)
- *VirtaMove, Corp. v. Google LLC*, Case No. 7:24-cv-00033 (W.D. Tex., Jan. 31, 2024)
- *VirtaMove, Corp. v. International Business Machines, Corp.*, Case No. 2:24-cv-00064 (E.D. Tex., Jan. 31, 2024)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of Rochette *et al.*	§	Attorney Docket No.: VIR058
	§	
U.S. Patent 7,784,058	§	
	§	
Issue Date: Aug. 24, 2010	§	Customer No.: 165774
	§	
Filing Date: Sep. 21, 2004	§	
	§	
For: COMPUTER SYSTEM HAVING USER MODE CRITICAL SYSTEM ELEMENTS AS SHARED LIBRARIES	§	

REQUEST FOR *EX PARTE* REEXAMINATION OF U.S. PATENT 7,784,058

Mail Stop "*Ex Parte* Reexam"
Attn: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Commissioner:

Pursuant to the provisions of 35 U.S.C. §§ 301-307, Unified Patents, LLC ("Requester") hereby requests an *ex parte* reexamination of claims 1 and 12 (the "Challenged Claims") of U.S. Patent 7,784,058 (the "'058 Patent, EX1001), which issued on August 24, 2010 to Donn Rochette et al. from U.S. Patent Application 10/946,536 (the "'536 Application"), filed on September 21, 2004, claiming priority to U.S. Provisional Application No. 60/504,213, filed on September 22, 2003.[1] The '058 Patent is currently assigned to VirtaMove, Corp (formerly known as Appzero Software Corp) ("VirtaMove" or "Patent Owner"). The assignment is recorded in the U.S. Patent and Trademark Office ("USPTO") at reel/frame 047157/0192.

Requester submits that this Request presents prior art references and analyses that are

[1] At this time, Requester assumes this priority date is correct.

noncumulative of the prior art that was before the Examiner during the original prosecution of the '058 Patent and that the Challenged Claims are invalid over these references. Requester therefore requests that an order for reexamination and an Office Action rejecting claims 1 and 12 be issued.

Ex Parte Patent Reexamination Filing Requirements

Pursuant to 37 C.F.R. § 1.510(b)(1), statements pointing out at least one substantial new question of patentability based on material, non-cumulative prior art for the Challenged Claims of the '058 Patent are provided in Section II of this Request.

Pursuant to 37 C.F.R. § 1.510(b)(2), reexamination of the Challenged Claims of the '058 Patent is requested, and a detailed explanation of the pertinence and manner of applying the cited references to the Challenged Claims is provided in Section III of this Request.

Pursuant to 37 C.F.R. § 1.510(b)(3), copies of every patent or printed publication relied upon or referred to in the statement pointing out each substantial new question of patentability or in the detailed explanation of the pertinence and manner of applying the cited references are provided as Exhibits 1001-1023 of this Request.

Pursuant to 37 C.F.R. §1.510(b)(4), a copy of the '058 Patent is provided as Exhibit 1001 of this Request, and with a copy of any disclaimer, certificate of correction, and reexamination certificate issued corresponding to the patent.

Pursuant to 37 C.F.R. § 1.510(b)(5), the attached Certificate of Service indicates that a copy of this Request, in its entirety, has been served on Patent Owner at the following address of record for Patent Owner, in accordance with 37 C.F.R. § 1.33(c):

27975 - ADD+G - 27975

1135 East State Road 434, Suite 3001

Winter Springs, FL 32708

Also submitted herewith is the fee set forth in 37 C.F.R. § 1.20(c)(1).

Pursuant to 37 C.F.R. § 1.510(b)(6), Requester hereby certifies that the statutory estoppel provisions of 35 U.S.C. § 315(e)(1) and 35 U.S.C. § 325(e)(1) do not prohibit Requester from filing this *ex parte* patent reexamination request.

TABLE OF CONTENTS

TABLE OF EXHIBITS

Exhibit	Description
Ex. 1001	U.S. Patent No. 7,784,058
Ex. 1002	File History for U.S. Patent No. 7,784,058
Ex. 1003	Declaration of Bruce McNair
Ex. 1004	Lucas, M. (2002). *Absolute BSD: the Ultimate Guide to FreeBSD,* No Starch Press ("*Lucas*").
Ex. 1005	Portions of the Internet Archive's captures of www.nostarch.com: Publisher's webpage for *Lucas*, last updated by the publisher on September 17, 2002, archived by Internet Archive on October 2, 2002, and available at https://web.archive.org/web/20021002074034/http://www.nostarch.com/abs_bsd.htm Absolute BSD Table of Contents webpage for *Lucas* last updated by the publisher on August 21, 2002, archived by Internet Archive on October 2, 2002, available at https://web.archive.org/web/20021002095939/http://www.nostarch.com/abs_bsd_toc.htm Reviews webpage for *Lucas* last updated by the publisher on August 22, 2002, archived by Internet Archive on December 24, 2002, available at https://web.archive.org/web/20021224224742/http://www.nostarch.com/abs_bsd_rev.htm Updates webpage for *Lucas* last updated by the publisher on August 21, 2002, archived by Internet Archive on December 24, 2002, available at https://web.archive.org/web/20021224231453/http://www.nostarch.com/abs_bsd_updates.htm
Ex. 1006	Amazon webpage for *Lucas*, archived by Internet Archive on September 13, 2002, available at https://web.archive.org/web/20020913131141/http://www.amazon.com:80/exec/obidos/tg/detail/-/1886411743/qid=1031807224/sr=1-5/?s=books&v=glance.
Ex. 1007	Library of Congress record on "Internet Archive," available at https://www.loc.gov/item/2003541624/.
Ex. 1008	Blott, S., Brustoloni, J., & Martin, C. (2000, March). NetTap: an efficient and reliable PC-based platform for network programming. In *2000 IEEE Third Conference on Open Architectures and Network Programming Proceedings*, pp. 13-22 ("*Blott*")
Ex. 1009	Copyright registration record *for 2000 IEEE third conference on open architectures and network programming: OPENARCH 2000* that contains *Blott*, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?Search_Arg=2000+IEEE+third+conference+on+open+architectures+and+network+programming&

	Search_Code=TALL&PID=Zp0gaXbixs2dnG5Dj4OAkZ5mIspYzL&SEQ=20240902151404&CNT=25&HIST=1 from the public catalog of the US Copyright Office.
Ex. 1010	Bibliographic and MARC records for *2000 IEEE Third Conference on Open Architectures and Network Programming Proceedings* that contains *Blott*, available at https://galileo-gatech.primo.exlibrisgroup.com/permalink/01GALI_GIT/18jodvt/alma9914167975702947 from the online catalog of the Georgia Tech Library; and OCLC FirstSearch record for this work, available on OCLC FirstSearch at https://firstsearch.oclc.org/ (subscription required for access).
Ex. 1011	Profile of Georgia Tech Library, available from https://librarytechnology.org/library/685 from libraries.org.
Ex. 1012	"FirstSearch Launched" *Library Systems Newsletter* (November 1991), available at https://librarytechnology.org/document/4992.
Ex. 1013	IEEE Metadata record for *Blott*, available at https://ieeexplore.ieee.org/document/828129 from the IEEE Xplore Digital Library.
Ex. 1014	IEEE Xplore home page archived by Internet Archive on June 22, 2000, available at https://web.archive.org/web/20000622050722/http://ieeexplore.ieee.org/lpdocs/epic03/newwelc.htm.
Ex. 1015	Kamp, P. H., & Watson, R. N. (2000, May). Jails: Confining the omnipotent root, in *Proceedings of the 2nd International SANE* Conference, 15 pages, available at http://www.sane.nl/events/sane2000/papers/kamp.pdf ("*Kamp*").
Ex. 1016	Portions of the Internet Archive's captures of SANE 2000 webpages: Home page of SANE 2000, the 2nd International SANE Conference last updated on February 4, 2000, archived by Internet Archive on December 20, 2005, available at https://web.archive.org/web/20051220143316/http://www.sane.nl/events/sane2000/sane2000-home.html Conference registration webpage last modified on February 1, 2000, archived by Internet Archive on May 19, 2006, available at https://web.archive.org/web/20060519212841/http://www.sane.nl/events/sane2000/registration/index.html Webpage for SANE 2000 papers last updated July 5, 2000, archived by Internet Archive on December 20, 2005, available at https://web.archive.org/web/20051220120838/http://www.sane.nl/events/sane2000/papers.html.
Ex. 1017	Archived copy of *Kamp* archived by Internet Archive on May 16, 2002, available at https://web.archive.org/web/20010516031656/www.docs.freebsd.org/44doc/papers/jail/jail.html
Ex. 1018	Curriculum Vitae of Ingrid Hsieh-Yee, Ph.D.
Ex. 1019	Understanding MARC Bibliographic, Parts VII to X, available at https://www.loc.gov/marc/umb/um07to10.html

Ex. 1020	Declaration of Dr. Hsieh-Yee
Ex. 1021	Curriculum Vitae of Bruce McNair
Ex. 1022	Bibliographic and MARC records for Lucas, from the online catalog of the Library of Congress, available at https://lccn.loc.gov/2002001428
Ex. 1023	The Library of Congress Online Catalog archived on March 31, 2001, by Internet Archive, available at https://web.archive.org/web/20010331110058/catalog.loc.gov/
Ex. 1024	Home page of SANE 2000, the 2nd International SANE Conference, last updated on February 4, 2000, archived by Internet Archive on August 17, 2000, available at https://web.archive.org/web/20000817214258/ http://www.nluug.nl/events/sane2000/sane2000-home.html
Ex. 1025	Conference registration webpage of SANE 2000, last modified on February 1, 2000, archived by Internet Archive on September 2, 2000, available at https://web.archive.org/web/20000902204901/ http://www.nluug.nl/events/sane2000/registration/index.html
Ex. 1026	Archived copy of *Kamp*, archived by Internet Archive on September 2, 2000, available at https://web.archive.org/web/20000902205700/ https://nluug.nl/events/sane2000/papers/kamp.pdf
Ex. 1027	Webpage for SANE 2000 papers last updated July 5, 2000, archived by Internet Archive on August 17, 2000, available at https://web.archive.org/web/20000817214322/ https://nluug.nl/events/sane2000/papers.html
Ex. 1028	U.S. Patent No. 8,601,580 to Hansen et al.
Ex. 1029	International Search Report for WO Publication No. 2003/058437 to Salomon

I. SUBSTANTIAL NEW QUESTIONS OF PATENTABILITY

Prior to describing the substantial new questions of patentability presented in this Request, provided below is an overview of the '058 Patent, a discussion of claim construction, and a summary of the prior art being discussed in the present Request.

A. U.S. Patent 7,784,058

1. Summary

The '058 Patent is titled "Computing system having user mode critical system elements as shared libraries," and is directed to a "computing system and architecture…that affects and extends services exported through application libraries." '058 Patent (EX1001), Abstract. More specifically, the '058 Patent contemplates a system that has an operating system and corresponding operating system kernel, where the kernel has "OS critical system elements (OSCSEs) for running in kernel mode." '058 Patent (EX1001), Abstract. The operating system also has "a shared library having critical system elements (SLCSEs) stored within the shared library for use by the software applications in user mode." '058 Patent (EX1001), Abstract.

Copies of the '058 Patent and its file history are provided as Exhibits 1001 and 1002, respectively. Claim 1, for which reexamination is requested, is reproduced below:

[1.0] A computing system for executing a plurality of software applications comprising:

[1.1] a) a processor;

[1.2] b) an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor; and,

[1.3] c) a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode and

[1.4] i) wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the

plurality of software applications,

[1.5] ii) wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function, and

[1.6] iii) wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously.

'058 Patent (EX1001), Claim 1.

In general, the claims for which reexamination is requested relate to systems that have "an operating system kernel having operating system critical system elements (OSCSEs)…to run in kernel mode," and "a shared library adapted to store replicas of at least some of the critical system elements, for use by the software applications in user mode executing in the context of the application." '058 Patent (EX1001), 1:58-64. The '058 Patent explains that a replica "is meant to denote a CSE having similar attributes to, but not necessarily and preferably not an exact copy of a CSE in the operating system (OS); notwithstanding, a CSE for use in user mode, may in a less preferred embodiment be a copy of a CSE in the OS." '058 Patent (EX1001), 1:66-2:3. Critical system elements include "[a]ny service or part of a service, 'normally' supplied by an operating system, that is critical to the operation of a software application." '058 Patent (EX1001), 6:6-8. The '058 Patent provides examples of critical system elements, such as "[n]etwork services including TCP/IP, Bluetooth, ATM; or message passing protocols." '058 Patent (EX1001), 6:11-13.

2. Prosecution History

The '058 Patent issued from U.S. Application No. 10/946,536, filed on September 21, 2004 (the "'536 Application"). The '536 Application claims priority to U.S. Provisional Application No. 60/504,213, filed on September 22, 2003.

A first non-final rejection issued on November 18, 2008, rejecting all claims under 35 U.S.C. § 112 as being indefinite, and rejecting all claims under § 102(b) as being anticipated by U.S. Patent No. 6,260,075 to Cabrero et al. A final rejection was issued on April 2, 2009, following amendments to the claims, maintaining rejections of all claims as being anticipated by Cabrero.

Following the final rejection, the applicant amended the claims to specify that the claimed "SLCSEs stored in the shared libraries are functional replicas of the OSCSEs." EX1002, 77 (emphasis in original). The applicant further argued that the prior art failed to disclose SLCSEs "being functional replicates of OSCSEs, or in other words, replacements," and failed to disclose SLCSEs being provided to one or more software applications without being shared with the other of the plurality of software applications. EX1002, 83-84.

A second non-final rejection issued on September 22, 2009, rejecting all claims under § 103 as obvious in view of U.S Patent No 6,212,574 to O'Rourke et al., U.S. Patent No. 5,481,706 to Peek, and U.S. Patent Application No. 2004/0025165 to Desoli et al. The applicant filed remarks and a request for reconsideration, arguing that the prior art was insufficient to disclose SLCSEs as functional replicas of OSCSEs, that the prior art was mischaracterized, and that a person of ordinary skill in the art would not combine the references. EX1002, 30-34. A notice of allowance then issued, where the examiner found that the prior art of record did not disclose the elements of claim 1. EX1002, 14-15.

This Request presents and applies new prior art with analysis that renders obvious every feature of the challenged claims of the '058 Patent. The prior art references cited in this Request were not applied by the examiner during prosecution. Thus, the prior art presented in this Request, along with the analysis presented by Mr. McNair (attached as Exhibit 1003), presents a substantial new question of patentability.

Requester therefore asks that an Order for Reexamination be issued and that the reexamination proceeding continue to reject and cancel claims 1 and 12.

B. Claim Construction

"During patent examination, the pending claims must be 'given their broadest reasonable interpretation consistent with the specification.'" MPEP § 2111. The '058 Patent has not yet expired; thus, the broadest reasonable interpretation standard applies. MPEP § 2258. The present Request presents the following claim analysis in a manner that is consistent with the broadest reasonable interpretation consistent with the specification.

C. Listing of Prior Art Patents and Printed Publications

Reexamination of the Challenged Claims is requested in view of the references listed below. Requester notes that each of the following references are printed publications. Requester includes the declaration of Dr. Hsieh-Yee (EX1020) supporting the dates that each of the references were publicly available.

- **Ex. 1004 ("Lucas")**: Lucas is a printed publication titled *Absolute BSD: the Ultimate Guide to FreeBSD*, authored by M. Lucas and published by No Starch Press in 2002. Lucas is prior art at least under 35 U.S.C. §§ 102(a), (b) because Lucas was publicly available by August 2002, more than one year before the September 22, 2003, priority date of the '058 Patent. *See* EX1020, ¶¶ 23-35 (citing EX1005; EX1006; EX1007).
- **Ex. 1008 ("Blott")**: Blott is a printed publication titled "NetTap: an efficient and reliable PC-based platform for network programming," authored by S. Blott et al. and published in *2000 IEEE Third Conference on Open Architectures and Network Programming* in March 2002. Blott is prior art at least under 35 U.S.C. §§ 102(a), (b) because Blott was included in conference proceedings published on March 26, 2002; appeared in an online library catalog on April 6, 2000; was physically publicly accessible in libraries starting on April 16, 2000; and was entered into IEEE Xplore on August 6, 2002, more than one year before the September 22, 2003, priority date of the '058 Patent. *See* EX1020, ¶¶ 36-58 (citing EX1009; EX1010; EX1011; EX1012; EX1013; EX1014).
- **Ex. 1015 ("Kamp")**: Kamp is a printed publication titled "Jails: Confining the omnipotent root," authored by P. H. Kamp et al. and was publicly accessible at the Sane 2000 conference in 2000. Kamp is prior art under at least 35 U.S.C. §§ 102(a), (b) because Kamp was provided to conference attendees at the May 2000 conference; was provided on the internet by August 17, 2000, via links; the corresponding text was archived on the FreeBSD documentation webpage on May 16, 2001; and was cited in papers published in 2001 and June 2002, all more

than one year before the September 22, 2003 priority date of the '058 Patent. *See* EX1020, ¶¶ 59-70 (citing EX1016; EX1017).

None of the references listed above were cited during prosecution of the '058 Patent. A Form SB-08 and copies of the cited references are submitted herewith. This request is also supported by the declaration of Bruce McNair (EX1003).

As shown below, Requester submits that the prior art references raise a new "substantial question of patentability" because "the teaching of the (prior art) patents and printed publications is such that a reasonable examiner would consider the teaching to be important in deciding whether or not the claim is patentable." *See* MPEP 2242. For example, the references discussed below and in further detail in the provided analysis, when considered as an ordered combination, teach each limitation of the Challenged Claims, including the idea of shared library critical system elements being functional replicas of kernel critical system elements. Further, they are new; these references were not previously considered; the "same question of patentability as to the claim has not been decided by the Office in an earlier concluded examination or review of the patent" at least because none of the art referenced in this request was before the Office during prosecution of the '058 Patent or during a prior post-grant proceeding challenging any claim of the '058 Patent, any of which that are known to Requester have been listed below in Section III, *infra*.

D. Lucas, Blott, and Kamp Qualify as Prior Art

As detailed further below, each of Lucas, Blott, and Kamp is prior art. "A reference is proven to be a 'printed publication' 'upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it.'" MPEP § 2128 (citing *In re Wyer,* 655 F.2d 221, 210 USPQ 790 (CCPA 1981) (quoting *I.C.E. Corp. v. Armco Steel Corp.,* 250 F. Supp. 738, 743, 148 USPQ 537, 540 (SDNY 1966))). Each of Lucas, Blott, and Kamp were made available to the extent persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, would have located it more than one year before the filing date of the '058 Patent.

1. Lucas is Prior Art

Lucas became prior art more than one year before the filing date of the '058 Patent. Specifically, Lucas was publicly accessible prior art at least as early as July 2002, and no later than

September 13, 2002, based on the following evidence:

- Exhibit 1004 is a complete, true and correct copy of *Absolute BSD: the Ultimate Guide to FreeBSD*, ("Lucas"), by M. Lucas, published by No Starch Press in 2002. Exhibit 1004 is a digitized copy produced from a physical copy of the book obtained from a book seller. The copyright page shows a 2002 copyright date. EX1020, ¶ 24.
- Internet Archive records of the website of No Starch Press, the publisher of Lucas, corroborates the copyright page of Lucas by reciting a publication date of July 2002 and reciting the same ISBN number. EX1005, 1; *see also* EX1020, ¶ 26. This webpage, archived on October 2, 2002, reflects a last update date of September 17, 2002, indicating that users would have been able to discover this webpage and purchase Lucas by September 17 and no later than October 2, 2002, on the Internet. EX1005, 1; *see also* EX1020, ¶ 27.
- Quotes regarding reviews of Lucas were published on the publisher's website, indicating that people of skill in the art would have been able to discover Lucas and Lucas had been read and reviewed by people of skill in the art. This webpage was archived on December 24, 2002, and reflects a last update date of August 22, 2002, indicating that people of skill in the art would have known and been able to obtain a copy of Lucas in August 2002. EX1005, 4-5; *see also* EX1020, ¶ 29.
- Internet Archive records of the Amazon.com webpage for Lucas further confirm that Lucas would have been available for purchase in August 2002. EX1006, 1-3; *see also* EX1020, ¶ 32-33. This webpage was archived on September 13, 2002, and reflects reviews dated as early as August 21, 2002, indicating that Lucas was publicly available and purchased as early as August 2002 and at least by September 13, 2002. EX1006, 2-3; *see also* EX1020, ¶ 33.
- Additionally, Lucas was catalogued in the Library of Congress as early as January 29, 2002, received at the Library of Congress as early as January 9, 2003, and would have been available to the public at the Library of Congress no later than May 2003. EX1022; EX1023; EX1020, ¶¶ 43, 45-46. Based on the bibliographic and MARC records for this library, interested individuals exercising reasonable diligence would have been able to discover Lucas using its ISBN number, its title, its author, its Library of Congress classification number (which is based on subject matter), its Dewey decimal number

(which is based on subject matter), and Library of Congress subject headings. EX1020, ¶ 42 (citing EX1022).

Thus, Lucas was on sale and otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could locate it, over a year before filing of the '058 Patent. And Lucas was further available in Libraries, with physical copies publicly accessible no later than May 2003, before the earliest effective priority date of September 22, 2003. Accordingly, Lucas qualifies as prior art.

2. Blott is Prior Art

Blott became prior art more than one year before the filing date of the '058 Patent. Specifically, Blott was publicly accessible prior art at least as early as March 26, 2000, and no later than August 6, 2002, based on the following evidence:

- Exhibit 1008 is a true and correct copy of "NetTap: an efficient and reliable PC-based platform for network programming," ("*Blott*"), by S. Blott, J. Brustoloni, and C. Martin, in *2000 IEEE Third Conference on Open Architectures and Network Programming*, Proceedings, pp. 13-22. Blott was obtained from the library of the Georgia Technical Institute. EX1020, ¶ 47. The copyright page shows a 2000 copyright date, an IEEE catalog number, the Library of Congress number, and the ISBN. EX1008, 3; *see also* EX1020, ¶ 48.
- The copyright registration record for the conference proceedings containing Blott reflect a date of publication of March 26, 2000, and a registration with the U.S. Copyright office of April 2, 2000. EX1009; *see also* EX1020, ¶¶ 49-50. The opening date of the conference was March 26, 2000. EX1008, 1.
- A copy of Blott was cataloged in the library of the Georgia Technical Institute as early as April 6, 2000, and a physical copy of Blott would have become accessible at this library no later than April 16, 2000. EX1010, 5; *see also* EX1020, ¶¶ 51-56.
- Blott was also cataloged in the IEEE Xplore Digital Library on August 6, 2002. EX1013, 1; *see also* EX1020, ¶¶ 63-66. Thus, a person of skill in the art would have been able to discover Blott through IEEE Xplore no later than August 6, 2002. Because IEEE has been a "well-known, reputable compiler and publisher of scientific and technical publications" the examiner may take "Official Notice that members in the scientific and

technical communities who both publish and engage in research rely on the information published on the copyright line of IEEE publications." *Ericsson, Inc. v. Intellectual Ventures I LLC*, IPR2014-00527, Paper No. 41, FWD at 10 (PTAB March 21, 2014) *aff'd under Rule 36*, No. 2015-1947 (Fed. Cir. August 16, 2016).

Thus, Blott was made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could locate it, over a year before filing of the '058 Patent.

3. Kamp is Prior Art

Kamp became prior art more than one year before the filing date of the '058 Patent. Specifically, Kamp was publicly accessible prior art as early as May 2000, and no later than August 17, 2000, based on the following evidence:

- Kamp (EX1026) is a true and correct copy of "Jails: Confining the omnipotent root," ("*Kamp*"), by P. H. Kamp and R. N. Watson, in *Proceedings of the 2nd International SANE Conference*. Kamp was obtained from https://web.archive.org/web/20000817214258/http://www.nluug.nl/events/sane2000/sane2000-home.html, an Internet Archive capture of the paper from August 17, 2000. This Internet Archive capture comes from the webpages for the 2000 SANE conference. *See, e.g.*, EX1027 (Webpage for SANE 2000 papers linking to the EX1024 capture of Kamp archived on August 17, 2000); EX1024 (Home page of SANE 2000, the 2nd International SANE Conference archived on August 17, 2000). Kamp sets forth a *prima facie* case that it was publicly accessible by August 17, 2000, capture by the Internet Archive's Wayback Machine. EX1026; *see* MPEP § 2128(II)(E) ("Prior art obtained via the Wayback Machine® sets forth a *prima facie* case that the art was publicly accessible at the date and time provided in the time stamp.").
- While the URLs archived are no longer live, currently active versions of the paper and related conference webpages can be found at http://www.sane.nl/events/sane2000/ and related URLs. *See* EX1015 (live webpage containing Kamp paper); EX1016 (live webpages for the 2000 SANE conference reflecting the same content as was archived in EX1025, EX1026, and EX1027). The accompanying declaration of Dr. Hsieh-Yee (EX1020) presents citations to the currently live webpages, but corresponding citations to

the Internet Archive versions that pre-date the priority date of the '058 Patent are provided in this Request.

- Kamp was presented at a conference that took place on May 22-25, 2000. EX1024, 1; EX1016, 1; EX1020, ¶ 73. Following the conference, the conference webpage linked to conference activities, including papers, at least by August 17, 2000, meaning interested users could follow the link to explore conference papers by this time. EX1027, 1 (Webpage for SANE 2000 papers linking to the EX1024 capture of Kamp archived on August 17, 2000); EX1016, 6; EX1020, ¶¶ 74-75. "Prior art disclosures on the Internet or on an online database are considered to be publicly available as of the date the item was publicly posted." MPEP § 2128(II)(B).
- A webpage version of Kamp was further available on the FreeBSD documentation webpage and archived as early as May 16, 2001. EX1017; EX1020, ¶¶ 77-79. The Internet Archive capture of the webpage version of Kamp (EX1017) was captured on May 16, 2001, and has substantially the same content as Kamp (EX1026). EX1020, ¶¶ 79-79. The webpage version of Kamp states that it was a "paper [that] was presented at the 2nd International System Administration and Networking Conference 'SANE 2000' May 22-25, 2000, in Maastricht, The Netherlands and is published in the proceedings." 14. While the Internet Archive version of Kamp does not display the image for Figure 1 in the capture, the webpage version of Kamp corroborates that the paper was published in the conference and that nearly all of its content was also available on the FreeBSD documentation website. EX1020, ¶ 78. This webpage also informs a person of skill in the art where to obtain Kamp, and it confirms that persons skilled in the art would have been aware of the paper from both the SANE conference and resources such as the FreeBSD documentation website.
- Kamp was cited by at least two other publications (a dissertation and a technical magazine) in October 2001 and June 2002, more than one year before the filing date of the '058 Patent. EX1020, ¶ 71.
- Kamp was publicly accessible to conference attendees in May 2000 at the SANE 2000 conference; was made publicly discoverable and accessible on the Internet in by the conference to interested users who could not attend the conference by July 2000, and no later than August 17, 2000; and was available on the FreeBSD documentation webpage.

EX1027, 1; EX1020, ¶¶ 80-81. Further, Kamp was accessed by individuals in that time period, as two papers published citing to Kamp in October 2001 and June 2002. EX1020, ¶ 81. And examiners have since cited Kamp during prosecution. EX1028, Cover at (56) (examiner citation to Kamp, "Jails: Confining the omnipotent root"); EX1029, 2 (international search report for WO 2003/058437 citing to "Jails: Confining the omnipotent root" by Kamp).

Thus, Kamp was made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could locate it, over a year before filing of the '058 Patent.

E. Overview of the Grounds Presenting Substantial New Questions of Patentability

The table below summarizes the grounds of unpatentability presented in this Request, all of which are governed by pre-AIA 35 U.S.C. § 103. Requester submits that each ground below raises a substantial new question (SNQ) of patentability for at least one claim of the '058 Patent.

Ground	Claim(s)	Prior Art
1	1 and 12	Lucas, Blott, and Kamp

F. Ground 1: Lucas in view of Blott and Kamp Presents New Questions of Patentability

1. Overview of Lucas

Lucas (EX1004) is a book titled *Absolute BSD: the Ultimate Guide to FreeBSD*, and provides "a one-stop shop" for information related to FreeBSD, an operating system that runs on a *computing system* such as a desktop or server implementing popular hardware options. "FreeBSD is a UNIX−like operating system, available freely over the Internet, that is used extensively in the ISP (Internet service provider) world, embedded devices, and anywhere reliability is paramount. It's based directly on the original UNIX produced by AT&T in the 1970s." Lucas (EX1004), XXX. FreeBSD "provide[s] a freely redistributable operating system that runs on popular hardware," such as"the Intel x86-compatible systems (386, 486, Pentium I through IV, Celeron, and AMD)." Lucas (EX1004), XXXV. Lucas provides information regarding the FreeBSD related to the development of the operating system, how to download and install the operating system, and

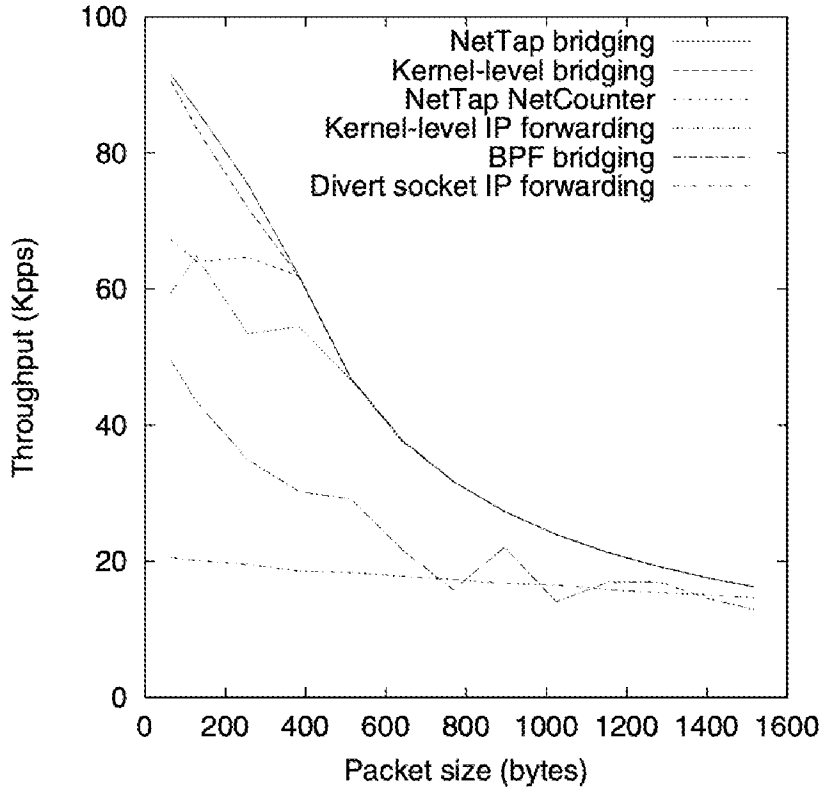
hardware setup for systems running FreeBSD. Lucas (EX1004), x-xi. The book then provides details regarding the kernel, networking, and other important system functions for systems running FreeBSD. Lucas (EX1004), xiii-xvi. One such feature of FreeBSD that Lucas describes is the implementation of FreeBSD jails. Lucas explains that FreeBSD includes functionality where "you can build an entire virtual machine on disk, and isolate that machine from the rest of your system. This is called a *jail*." Lucas (EX1004), 175. These jails provide isolated environments where applications can run in a restricted environment, without access to anything outside of the jail. *See* Lucas (EX1004), 175; EX1003, ¶ 40.

Lucas is analogous art to the '058 Patent. Like the '058 patent, Lucas is in the field of computer systems, and more specifically to an architecture that affects and extends services exported through application libraries. *See, e.g.*, Lucas (EX1004), 260 (describing application libraries in FreeBSD and explaining that "[t]he basic idea behind a shared library is quite straightforward: It's a chunk of compiled code that provides services and functions to other chunks of compiled code. Shared libraries provide popular functions for all programs to use, and they are designed to be reused by as many different programs as possible."). Further, Lucas is reasonably pertinent to at least one problem concerning the inventor of the '058 Patent (the "ability to deploy multiple instances of a CSE", '058 Patent, 1:48-50) because it provides a mechanism by which applications are run in isolated environments with their own set of system elements, such as system libraries. *See infra* Section II.A.1.[1.3]-[1.4]. EX1003, ¶¶ 41-42.

2. Overview of Blott

Blott (EX1008) is a printed publication titled NetTap: an efficient and reliable PC-based platform for network programming," and discloses "a new platform for prototyping, field- testing, and deploying network services." Blott's platform is based on a computer running FreeBSD, and specifically describes a set of libraries which provide "substantial performance advantages" over FreeBSD kernel level implementations of IP forwarding. Specifically, Blott "provides user-level libraries for TCP/IP protocols, IP security [16], and output link scheduling [7]," which it refers to as NetTap. Blott (EX1008), 6. Blott explains that NetTap provides IP forwarding for applications at the user level instead of the kernel level. *See* Blott (EX1008), 1 ("In particular, NetCounter, an application for capturing and aggregating IP network billing records, achieves greater throughput on NetTap at user level than does FreeBSD's IP forwarding at kernel level."), 3 ("NetTap

supports…network applications at user level, avoiding the difficulties associated with kernel-level software."). Blott also explains that NetTapp provides bridging for applications at the user level instead of the Kernel level. *See* Blott (EX1008), 8 ("NetTap's user-level bridging slightly outperformed kernel-level bridging because both NetTap and kernel-level bridging run essentially without copying or system call overheads, but NetTap has the advantage that its queueing of incoming packets helps avoid the onset of packet loss.… For the NetCounter application and 64-byte packets, NetTap provided a throughput of about 67 Kpps, more than the throughput of kernel-level IP forwarding (59 Kpps)"). Blott provides comparisons showing the performance differences between its NetTap implementations of functions such as bridging and IP forwarding, as compared to kernel-level bridging and forwarding. As reflected in the chart below, Blott's NetTap implementation provides performance advantages for each:



Blott (EX1008), 8; EX1003, ¶ 43.

Blott is analogous art to the '058 Patent. Blott is within the field of endeavor of the '058 Patent ("computer systems, and more specifically to an architecture that affects and extends services exported through application libraries") because it relates to a FreeBSD architecture and

providing APIs with a performance advantage over kernel-level implementations in the FreeBSD operating system for use with network applications. Blott (EX1008), 1. Further, Blott is reasonably pertinent to at least one problem concerning the inventor of the '058 Patent (functional replication of kernel-level critical system elements at the user-level) because Blott details an approach to provide functional and advantageous user-level implementations of functions such as IP forwarding and bridging, which are usually performed at the kernel-level by the FreeBSD operating system. *See* Blott (EX1008), 1, 3, 8; EX1003, ¶¶ 44-45.

3. Overview of Kamp

Kamp (EX1026) is a printed publication titled "Jails: Confining the omnipotent root," and provides documentation detailing the implementation of jails within the FreeBSD operating system. For instance, an example of a FreeBSD system with two configured jails is proved in the schematic diagram below:

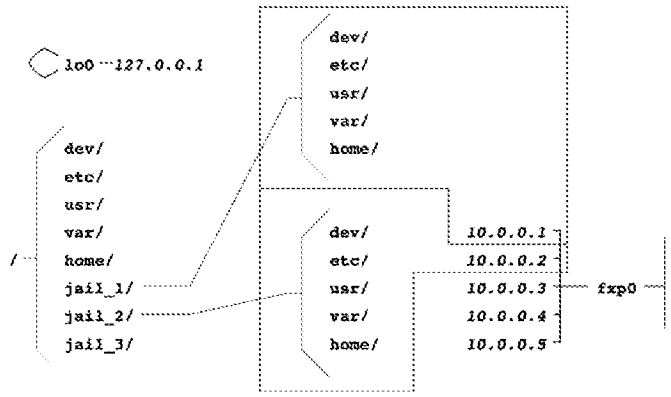


Fig. 1 — Schematic diagram of machine with two configured jails

Kamp (EX1026), 5. Jails within the FreeBSD operating system "creates a complete FreeBSD installation for each jail," including copies of system binaries and data files. Kamp (EX1026), 9. "Rather than introduce additional fine-grained access control mechanism, we partition a FreeBSD environment (processes, file system, network resources) into a management environment, and optionally subset Jail environments. In doing so, we simultaneously maintain the existing UNIX security model, allowing multiple users and a privileged root user in each jail, while limiting the scope of root's activities to his jail." Kamp (EX1026), 4-5. Kamp explains that when jails are created, activities of a user and processes running within a jail are restricted to their respective jail, such that processes may not leave the jail or otherwise manipulate system resources outside of the

jail. *See* Kamp (EX1026), 4-5; EX1003, ¶ 46.

In the example provided in Figure 1, above, Kamp shows two configured jails (jail_1 and jail_2) running on a FreeBSD system. Kamp (EX1026), 5. Each of jail_1 and jail_2 is provided a copy of directories such as /dev, /etc, /usr, /var, and /home, as each jail is "a complete FreeBSD installation" which includes copies of all relevant system binaries, data files, and its own /etc directory." Kamp (EX1026), 9; EX1003, ¶ 47.

Kamp is analogous art to the '058 Patent. Like the '058 patent, Kamp is in the field of computer systems, and more specifically to an architecture that affects and extends services exported through application libraries. *See, e.g.,* Kamp (EX1026), 9 (explaining that jails extend "copies of all relevant system binaries, data files," and directories in creating "a complete FreeBSD installation for each jail"). Further, Kamp is reasonably pertinent to at least one problem concerning the inventor of the '058 Patent (the "ability to deploy multiple instances of a CSE", '058 Patent, 1:48-50) because it provides further details regarding FreeBSD jails, a mechanism by which applications are run in isolated environments with their own set of system elements, such as system libraries. *See* Kamp (EX1026), 9; *see also infra* Section II.A.1.[1.3]-[1.4]. EX1003, ¶¶ 48-49.

G. Motivation To Combine Lucas, Blott, and Kamp

As an initial matter, a POSITA would have been motivated to combine the teachings of each of Lucas, Blott, and Kamp, as each of the three references focuses on different implementation details of the same operating system: FreeBSD. Lucas provides a high-level overview of FreeBSD and some details regarding the implementation of the operating system, such as the hardware that would be used, how the operating system would be set up and configured, and features that exist in the operating system as detailed further in the documentation for FreeBSD. Kamp provides additional details focused to one element of FreeBSD, jails, which are discussed in Lucas and were added to the FreeBSD operating system and corresponding documentation. Indeed, Lucas can be found on the FreeBSD documentation webpages, as reflected in Internet Archive captures from before the priority date of the '058 Patent. *See* EX1016. And Blott provides improvements specifically focused on the FreeBSD operating system. In this way, each of Lucas, Blott, and Kamp provide different perspectives on the same operating system, FreeBSD, which POSITAs were familiar with and used. Accordingly, a POSITA would have been motived to

combine each of the references due to a teaching, suggestion, or motivation in the prior art, as each piece of prior art is directed to FreeBSD, and would have suggested to a POSITA that the teachings of each of the references are compatible. *See* MPEP § 2143(G). Further, both Lucas and Kamp provide details regarding jails within FreeBSD, and in the combination, a POSITA would simply look to Kamp for additional information regarding the operation of jails in a FreeBSD operating system, which operates as described by Lucas. And as Mr. McNair notes, because the teachings provided in the references are the "basic presentation of good engineering design to solve practical problems, that is the approach a POSITA would follow to solve the same problem." EX1003, ¶ 50.

Further, a POSITA would have been motivated to combine Lucas and Kamp with Blott, and doing so would have been the use of a known technique to improve similar devices in the same way. The combination of Lucas and Kamp is focused on the FreeBSD operating system, and the implantation of jails within such systems. Blott further relates to FreeBSD, by providing a platform directed to devices running FreeBSD. Blott (EX1008), 1. More specifically, Blott provides an API for the FreeBSD operating system to provide functionality such as bridging and IP forwarding. Blott (EX1008), 1. In the combination, a machine running FreeBSD, as detailed by Lucas and Kamp, would further implement the NetTap API detailed by Blott. A POSITA would have been motivated to incorporate Blott's NetTap API in a FreeBSD system because Blott provides experimental evidence showing "substantial performance advantages" over the kernel-level equivalent functionality provided in FreeBSD. Blott (EX1008), 1, 7-9. Further, a POSITA would have had a reasonable expectation of success in combining Blott with the Lucas-Kamp combination because each of Blott, Lucas, and Kamp are focused on the FreeBSD operating system, and Blott provides experimental evidence showing success of its API running on FreeBSD. Blott (EX1008), 1, 7-9; EX1003, ¶ 51.

II. DETAILED APPLICATION OF THE PRIOR ART TO EVERY CLAIM FOR WHICH REEXAMINATION IS REQUESTED

A. Ground 1: The combination of Lucas, Blott and Kamp renders claims 1 and 12 obvious

1. Claim 1

[1.0][2] A computing system for executing a plurality of software applications comprising:

To the extent it is limiting, Lucas renders obvious *a computing system* (hardware systems running the FreeBSD operating system) *for executing a plurality of software applications* (a desktop machine or server environment for use with web, mail, file, and support services and corresponding applications). EX1003, ¶¶ 53-56.

Lucas provides "a one-stop shop" for information related to FreeBSD, an operating system that runs on a *computing system* such as a desktop or server implementing popular hardware options. "FreeBSD is a UNIX−like operating system, available freely over the Internet, that is used extensively in the ISP (Internet service provider) world, embedded devices, and anywhere reliability is paramount. It's based directly on the original UNIX produced by AT&T in the 1970s." Lucas (EX1004), XXX. FreeBSD "provide[s] a freely redistributable operating system that runs on popular hardware." Lucas (EX1004), XXXV. Such *computing systems* contemplated by Lucas include "the Intel x86-compatible systems (386, 486, Pentium I through IV, Celeron, and AMD)." Lucas (EX1004), XXXV. EX1003, ¶ 54.

Hardware systems running the FreeBSD operating system *execut[e] a plurality of software applications* on both desktop and server environments. Lucas explains that "While FreeBSD can be used as a very **powerful desktop or development machine**, its history shows a strong bias toward **Web, mail, file, and support services**. In fact, FreeBSD's main strength is on Internet servers, and it is an excellent choice for any Internet service." A variety of applications may be run on hardware systems running the FreeBSD operating system (i.e., systems *execute a plurality of software applications*) including various web browsers (Netscape, Mozilla), GUI mail readers

[2] A preamble is "'generally not limiting,'" and should not be considered so for claim 1. *Symantec Corp. v. Computer Assoc. Intern.*, 522 F.3d 1279 (Fed. Cir. 2008) (quoting *Catalina Mktg. Int'l, Inc. v. Coolsavings.com, Inc.*, 289 F.3d 801, 808 (Fed. Cir. 2002)). The body of claim 1 provides a structurally complete invention, while the preamble simply provides context for the claimed device for calculating a current load level. *Id.*, 522 F.3d at 1289.

(Mutt, Pine), office suites (StarOffice, OpenOffice, Koffice), Music, Graphics, Desk Utilities (GnuCash, Palm), and games (Xevil, Heretic, Doom, Quake, etc.) which are installed and executed. Lucas (EX1004), 510-515. EX1003, ¶ 55.

Thus, Lucas renders obvious *a computing system* (hardware systems running the FreeBSD operating system) *for executing a plurality of software applications* (a desktop machine or server environment for use with web, mail, file, and support services and corresponding applications). EX1003, ¶¶ 53-56.

[1.1] a) a processor;

Lucas renders obvious *a processor* (hardware systems running the FreeBSD operating system use processors such as Intel 386, 486, Pentium I through IV, Celeron, and AMD). EX1003, ¶¶ 57-59.

As noted above in the discussion of element [1.0], Lucas contemplates computing systems that run on hardware including include "Intel x86-compatible systems (386, 486, Pentium I through IV, Celeron, and AMD)." Lucas (EX1004), XXXV. These systems run on *a processor* such as an Intel 386, 486, Pentium, Celeron, or AMD processor. Indeed, "FreeBSD's goal is to provide a freely redistributable operating system that runs on popular hardware. While system security is a vital concern, **FreeBSD's main goal is to run on the hardware people are most likely to have. Today, this means the Intel x86−compatible systems (386, 486, Pentium I through IV, Celeron, and AMD)**. FreeBSD also supports the Alpha processor, and work is underway to support Intel's new IA64, AMD's new 64−bit chips, and Motorola's PowerPC, as well as Sun's SPARC. (These platforms aren't afterthoughts; the hardware is just now coming out, or only now becoming popular enough to port to.)." Lucas (EX1004), XXXV. EX1003, ¶ 58.

Thus, Lucas renders obvious *a processor* (hardware systems running the FreeBSD operating system use processors such as Intel 386, 486, Pentium I through IV, Celeron, and AMD). EX1003, ¶¶ 57-59.

[1.2] b) an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor; and,

Lucas in view of Blott renders obvious *an operating system* (FreeBSD) *having an operating system kernel* (the FreeBSD operating system contains a kernel) *having OS critical system elements (OSCSEs)* (networking services such as IP packet forwarding and bridging) *for*

running in kernel mode (networking services such as IP packet forwarding and bridging are run at the kernel level) *using said processor* (FreeBSD performs kernel-level IP packet forwarding and bridging on systems running on Intel processors). EX1003, ¶¶ 60-65.

As noted above in the discussion of element [1.0], Lucas provides "a one-stop shop" for information related to FreeBSD, an operating system that runs on a computing system such as a desktop or server implementing popular hardware options. "**FreeBSD is a UNIX−like operating system**, available freely over the Internet, that is used extensively in the ISP (Internet service provider) world, embedded devices, and anywhere reliability is paramount. It's based directly on the original UNIX produced by AT&T in the 1970s." Lucas (EX1004), XXX. EX1003, ¶ 61.

The '058 Patent defines a critical system element as "[a]ny service or part of a service, 'normally' supplied by an operating system, that is critical to the operation of a software application." '058 Patent, 6:6-8. The '058 Patent provides examples of critical system elements, including "[n]etwork services including TCP/IP, Bluetooth, ATM; or message passing protocols." '058 Patent, 6:11-13. Indeed, the '058 Patent acknowledges that the TCP/IP stack of BSD is a critical system element: "Applications on a Linux platform may use a **BSD TPC/IP stack in the form of a CSE**." '058 Patent, 5:46-47. EX1003, ¶ 62.

The FreeBSD operating system contains a kernel (*an operating system having an operating system kernel*), which is configured when setting up the operating system: "The first step in optimizing FreeBSD is to configure the kernel." Lucas (EX1004), 61. As confirmed by Blott, the FreeBSD operating system kernel includes *OS critical system elements (OSCSEs)* related to networking, such as packet forwarding and bridging. For instance, Blott explains that FreeBSD's "ipfw API supports kernel-level firewalling applications in router or bridge configurations." Blott (EX1008), 4. Blott also acknowledges that FreeBSD performs "IP forwarding at the kernel level." Blott (EX1008), 1. Indeed, Blott provides alternatives to kernel-level bridging and IP forwarding. *See, e.g.,* Blott (EX1008), 8. EX1003, ¶ 63.

Further, as noted in the analysis of element [1.1] above, hardware systems running the FreeBSD operating system use Intel processors. *See* Lucas (EX1004), XXXV. Thus, the FreeBSD kernel processes described in Blott, such as IP packet forwarding and bridging (*OS critical system elements*) are executed at the kernel level (*running in kernel mode*) on Intel processors that are running FreeBSD (*running in kernel mode using said processor*). EX1003, ¶ 64.

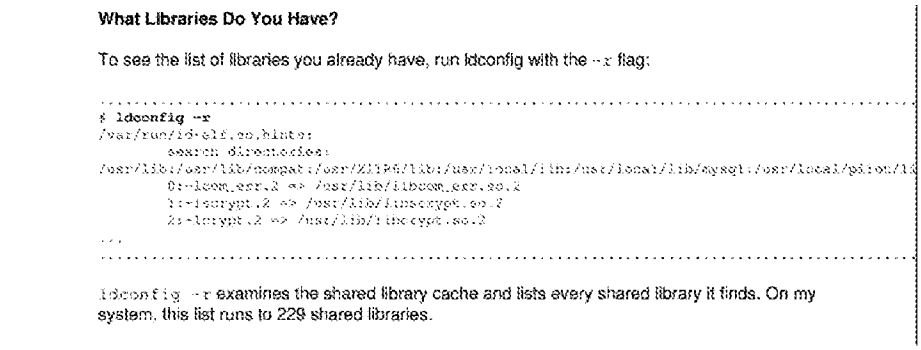
Thus, Lucas in view of Blott renders obvious *an operating system* (FreeBSD) *having an operating system kernel* (the FreeBSD operating system contains a kernel) *having OS critical system elements (OSCSEs)* (networking services such as IP packet forwarding and bridging) *for running in kernel mode using said processor* (FreeBSD performs kernel-level IP packet forwarding and bridging on systems running on Intel processors). EX1003, ¶¶ 60-65.

[1.3] c) a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode and

Lucas renders obvious *a shared library* (the /usr directory of the FreeBSD file system) *having shared library critical system elements (SLCSEs)* (shared libraries) *stored therein* (the /usr directory of the FreeBSD file system stores shared libraries, which provide common functions for software) *for use by the plurality of software applications in user mode* (programs access shared libraries when running). EX1003, ¶¶ 66-70.

As noted above in the discussion of element [1.2], the '058 Patent defines a critical system element as "[a]ny service or part of a service, 'normally' supplied by an operating system, that is critical to the operation of a software application." '058 Patent, 6:6-8. The '058 Patent provides examples of critical system elements, including "[n]etwork services including TCP/IP, Bluetooth, ATM; or message passing protocols." '058 Patent, 6:11-13. The '058 Patent further explains that critical system elements are *shared library critical system elements (SLCSEs)*, for example, when critical system element such as TCP/IP are provided in user mode. '058 Patent, 5:46-47. EX1003, ¶ 67.

Lucas explains that FreeBSD has a collection of shared libraries, which are cached and provided to programs to provide common functions (i.e., shared libraries provide *shared library critical system elements (SLCSEs)*): "The basic idea behind a shared library is quite straightforward: It's a chunk of compiled code that provides services and functions to other chunks of compiled code. Shared libraries provide popular functions for all programs to use, and they are designed to be reused by as many different programs as possible." Lucas (EX1004), 260. These libraries are found with the /usr directory (*a shared library* storing FreeBSD's collection of shared libraries) in the FreeBSD file system, and reflected within a cache of shared libraries that is built at boottime by the operating system. Lucas (EX1004), 260-261. Lucas explains that this is reflected in the FreeBSD operating system when "ldconfig" is run with the -r flag:



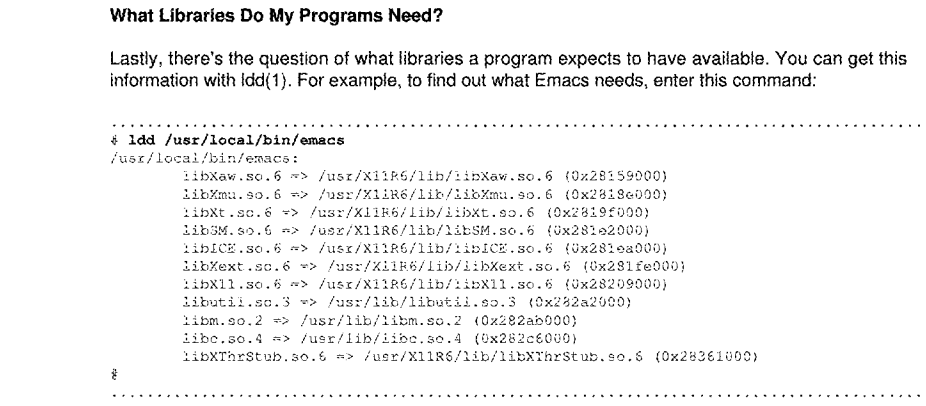
What Libraries Do You Have?

To see the list of libraries you already have, run ldconfig with the -r flag:

```
# ldconfig -r
/var/run/ld-elf.so.hints:
        search directories:
/usr/lib:/usr/lib/compat:/usr/X11R6/lib:/usr/local/lib:/usr/local/lib/mysql:/usr/local/pgsql/lib
        0:-lcom_err.2 => /usr/lib/libcom_err.so.2
        1:-lcrypt.2 => /usr/lib/libcrypt.so.2
        2:-lcrypt.2 => /usr/lib/libcrypt.so.2
...
```

ldconfig -r examines the shared library cache and lists every shared library it finds. On my system, this list runs to 229 shared libraries.

Lucas (EX1004), 261 (showing libraries found in "/usr/lib" in response to a user running "ldconfig -r"). Lucas also explains that the "ldd" command can be used to see what libraries a program expects to have available when the program is run:



What Libraries Do My Programs Need?

Lastly, there's the question of what libraries a program expects to have available. You can get this information with ldd(1). For example, to find out what Emacs needs, enter this command:

```
# ldd /usr/local/bin/emacs
/usr/local/bin/emacs:
        libXaw.so.6 => /usr/X11R6/lib/libXaw.so.6 (0x28159000)
        libXmu.so.6 => /usr/X11R6/lib/libXmu.so.6 (0x2818e000)
        libXt.so.6 => /usr/X11R6/lib/libXt.so.6 (0x2819f000)
        libSM.so.6 => /usr/X11R6/lib/libSM.so.6 (0x281e2000)
        libICE.so.6 => /usr/X11R6/lib/libICE.so.6 (0x281ea000)
        libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0x281fe000)
        libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x28209000)
        libutil.so.3 => /usr/lib/libutil.so.3 (0x282a2000)
        libm.so.2 => /usr/lib/libm.so.2 (0x282ab000)
        libc.so.4 => /usr/lib/libc.so.4 (0x282c6000)
        libXThrStub.so.6 => /usr/X11R6/lib/libXThrStub.so.6 (0x28361000)
#
```

Lucas (EX1004), 263. EX1003, ¶ 68.

The /usr directory of the FreeBSD file system stores shared libraries that are accessed by a number of programs to provide common functionality (i.e., the stored shared libraries are *shared library critical system elements* because programs rely on the shared libraries in order to run): "For example, many programs must hash (or one-way encrypt) data as part of their function. But if every program had to include hashing code, each would be larger, harder to write, and more unpleasant to maintain. What's more, programs would have interoperability problems if they implemented hashes differently. By using a shared library (in this example, libcrypt), a program that needs hashing has access to the functions while eliminating problems of maintenance and interoperability. Similarly, other shared libraries provide common functions to support other software. This reduces the average size of programs, freeing up a reasonably large amount of system memory." Lucas (EX1004), 260. It would have been obvious to a POSITA that the /usr directory and corresponding shared libraries are *for use by the plurality of software applications*

in user mode because programs access these shared libraries when running. EX1003, ¶ 69.

Thus, Lucas renders obvious *a shared library* (the /usr directory of the FreeBSD file system) *having shared library critical system elements (SLCSEs) stored therein* (the /usr directory of the FreeBSD file system stores shared libraries, which provide common functions for software) *for use by the plurality of software applications in user mode* (programs access shared libraries when running).

[1.4] wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications,

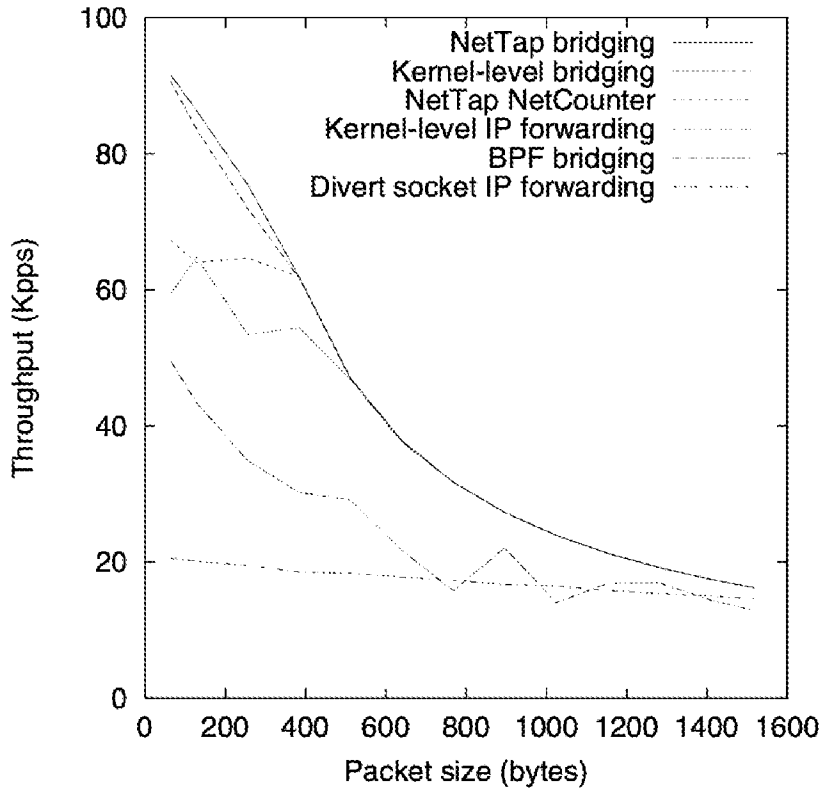
Lucas in view of Blott renders obvious *wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs* (replicas of FreeBSD kernel level functions, such as IP forwarding, can be provided through user-level implementations, such as Blott's NetTap, stored in the /usr directory) *and are accessible to some of the plurality of software applications* (programs can access user-level NetTap implementations of FreeBSD kernel-level functions such as IP forwarding) *and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications* (applications run directly on top of user-level NetTap implementations of FreeBSD kernel-level functions such as IP forwarding). EX1003, ¶¶ 71-76.

As noted above in the analysis of element [1.3], Lucas explains that the FreeBSD operating system stores shared libraries in the /usr directory (*SLCSEs stored in the shared library*). Blott describes a FreeBSD platform that can be added to the FreeBSD operating system which provides functions at the user-level replicating kernel-level critical system elements. EX1003, ¶ 72.

Blott provides "a new platform for prototyping, field- testing, and deploying network services." Blott's platform is based on a computer running FreeBSD, and specifically describes a set of libraries which provide "substantial performance advantages" over FreeBSD kernel level implementations of IP forwarding. Specifically, Blott "provides user-level libraries for TCP/IP protocols, IP security [16], and output link scheduling [7]," which it refers to as NetTap (e.g., *SLCSEs*). Blott, 6. The NetTap APIs (*SLCSEs*) provide user-level bridging and user-level IP forwarding within its user-level libraries (*SLCSEs*) as alternatives to kernel-level implementations (*OSCSEs for running in kernel mode*, e.g., networking services such as IP packet forwarding and

bridging as discussed above in the analysis of element [1.2]). EX1003, ¶ 73.

Blott explains that NetTap (*SLCSEs*) provides processing for network applications at the user level instead of the kernel level (e.g., NetTap provides *functional replicas of OSCSEs* such as FreeBSD kernel level functions for IP forwarding at the user level). *See, e.g.*, Blott (EX1008), 1 ("In particular, NetCounter, an application for capturing and aggregating IP network billing records, achieves greater throughput on NetTap at user level than does FreeBSD's IP forwarding at kernel level."), 3 ("NetTap supports…network applications at user level, avoiding the difficulties associated with kernel-level software."), 8 ("NetTap's **user-level bridging slightly outperformed kernel-level bridging** because both NetTap and kernel-level bridging run essentially without copying or system call overheads, but NetTap has the advantage that its queueing of incoming packets helps avoid the onset of packet loss.… For the NetCounter application and 64-byte packets, **NetTap provided a throughput of about 67 Kpps, more than the throughput of kernel-level IP forwarding** (59 Kpps)"). This is similarly reflected in the charts shown in Blott, comparing the performance of NetTap at the user level to the functional equivalents at the kernel level. More specifically, the chart shows a comparison of throughput for packets of different sizes using NetTap bridging vs Kernel-level bridging, showing minor advantages to NetTap bridging at small packet sizes, and NetTap NetCounter for IP forwarding vs Kernel-level IP forwarding, showing advantages for NetTap NetCounter at smaller packet sizes:



Blott (EX1008), 8. EX1003, ¶ 74.

The NetTap libraries are available to applications that are "written in any language and implement arbitrary functionality." Blott (EX1008), 6. "These libraries run on top of the NetTap API. Applications may easily specialize or modify NetTap's libraries." Blott (EX1008), 6. Thus, applications run directly on top of user-level NetTap implementations of FreeBSD kernel-level functions such as IP forwarding. EX1003, ¶ 75.

Thus, Lucas in view of Blott renders obvious *wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs* (replicas of FreeBSD kernel level functions such as IP forwarding can be provided through user-level implementations, such as Blott's NetTap, stored in the /usr directory) *and are accessible to some of the plurality of software applications* (programs can access user-level NetTap implementations of FreeBSD kernel-level functions such as IP forwarding) *and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications* (applications run directly on top of user-level NetTap implementations of FreeBSD kernel-level functions such as IP forwarding). EX1003, ¶¶ 71-76.

[1.5] ii) wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function, and

Lucas in view of Blott and Kamp renders obvious *wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library* (a copy of the libraries, processes, file system, and network resources within FreeBSD, including NetTap, are provided to FreeBSD jail_1) *is run in a context of said at least first of the plurality of software applications* (applications operating in a FreeBSD jail_1 use jail_1's libraries, processes, file system, and network resources, including NetTap) *without being shared with other of the plurality of software applications* (i.e., software operating outside of FreeBSD jail_1 cannot access the libraries, processes, file system, and network resources inside jail_1, including NetTap) *and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function* (FreeBSD jail_2 can be run with a unique copy of the libraries, processes, file system, and network resources, including NetTap, used by applications running in jail_2). EX1003, ¶¶ 77-88.

Lucas explains that FreeBSD includes functionality where "you can build an entire virtual machine on disk, and isolate that machine from the rest of your system. This is called a *jail*." Lucas (EX1004), 175. Kamp provides further documentation detailing the implementation of jails within the FreeBSD operating system, and provides a schematic diagram of a machine with two configured jails:

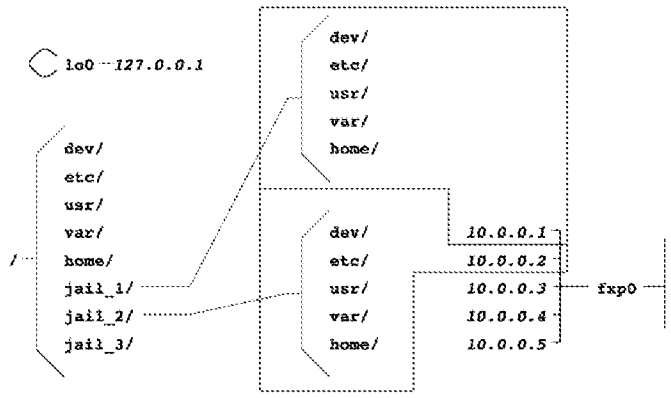


Fig. 1 — Schematic diagram of machine with two configured jails

Kamp (EX1026), 5. EX1003, ¶ 78.

When jails are initially set up within an operating instance of FreeBSD, the FreeBSD jail configuration "creates a complete FreeBSD installation for each jail. This includes copies of all relevant system binaries, data files, and its own /etc directory." Kamp (EX1026), 9. A FreeBSD jail operates as an instance within the FreeBSD operating system, with its own processes, file system, and network resources available only to the jail: "Rather than introduce additional fine-grained access control mechanism, we partition a FreeBSD environment (processes, file system, network resources) into a management environment, and optionally subset Jail environments. In doing so, we simultaneously maintain the existing UNIX security model, allowing multiple users and a privileged root user in each jail, while limiting the scope of root's activities to his jail." Kamp (EX1026), 4-5. These jails are instances of "a complete copy of FreeBSD" running within the operating system. *See* Lucas (EX1004), 177. This is reflected in Fig. 1 of Kamp, where two configured jails are shown, each having a copy of the directories such as /dev, /etc, /usr, /var, and /home:

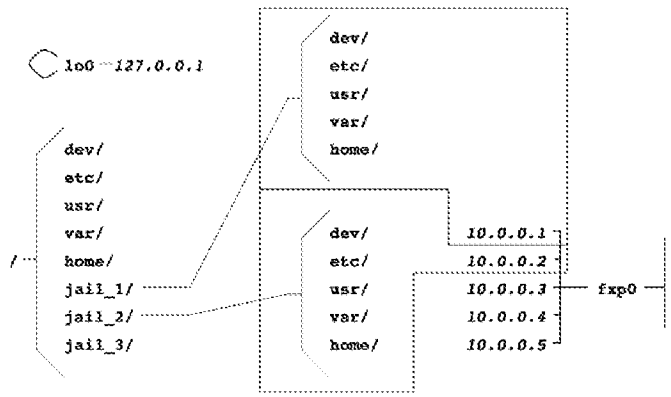


Fig. 1 — Schematic diagram of machine with two configured jails

Kamp (EX1026), 5. EX1003, ¶79.

Thus, in the combination, a FreeBSD jail created on a system implementing Blott's NetTap would include a copy of NetTap, stored in the jail's /usr directory created with the jail. In a machine with two configured jails, as shown by Kamp, a copy of the libraries, processes, file system, and network resources within FreeBSD, including NetTap, are provided to FreeBSD jail_1 (*an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library*). EX1003, ¶ 80.

Processes running within a jail may have full access to the libraries, processes, file system, and network resources within the jail (i.e., *provided to at least a first of the plurality of software applications from the shared library*). EX1003, ¶ 81.

"When in jail, clients can have root access and even install whatever nifty toys they desire without interfering with the main system. All processes that are running in the jail are restricted to the jail environment, and the kernel does not give them access to any information not in their jail. The filesystem in the jail does not know about files or filesystems outside the jail. Since no program or process in the jail knows about anything outside the jail, and cannot read or access anything outside the jail, the user is locked in. Not only can the client not break out of the jail, if the jail is hacked the intruder can't break out of the jail. This helps secure your system while meeting client needs." Lucas (EX1004), 176. EX1003, ¶ 82.

In a machine with two configured jails, as shown by Kamp, software running in jail_1, below, is provided a copy of directories such as /dev, /etc, /usr, /var, and /home, as shown below. As discussed above in the analysis of element [1.3], /usr includes system libraries, and as discussed above in the analysis of element [1.4], in the combination /usr would include the libraries provided by NetTap.

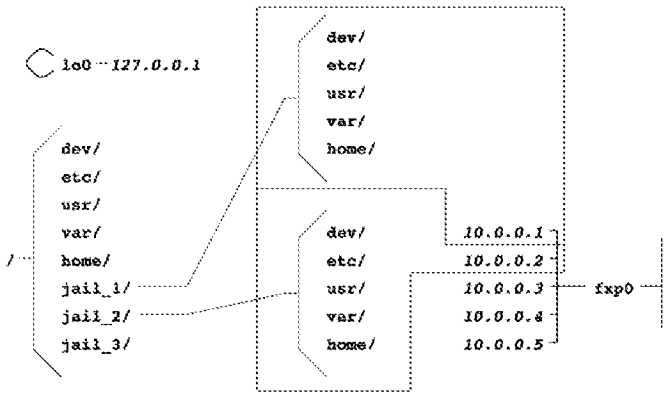


Fig. 1 — Schematic diagram of machine with two configured jails

Kamp (EX1026), 5. EX1003, ¶ 83.

These directories and corresponding libraries, processes, file system, and network resources are not shared with any software operating outside of jail_1 (*without being shared with other of the plurality of software applications*): "When in jail, clients can have root access and even install whatever nifty toys they desire without interfering with the main system. All processes

that are running in the jail are restricted to the jail environment, and the kernel does not give them access to any information not in their jail. The filesystem in the jail does not know about files or filesystems outside the jail. Since no program or process in the jail knows about anything outside the jail, and cannot read or access anything outside the jail, the user is locked in. Not only can the client not break out of the jail, if the jail is hacked the intruder can't break out of the jail. This helps secure your system while meeting client needs." Lucas (EX1004), 175. EX1003, ¶ 84.

In a machine with two configured jails, as shown by Kamp, software running in jail_2, below, (*a second of the plurality of software applications*) runs on a second jail, separate from jail_1, with its own a copy of directories such as /dev, /etc, /usr, /var, and /home (i.e., runs using *a unique instance of a corresponding critical system element for performing the same function*)

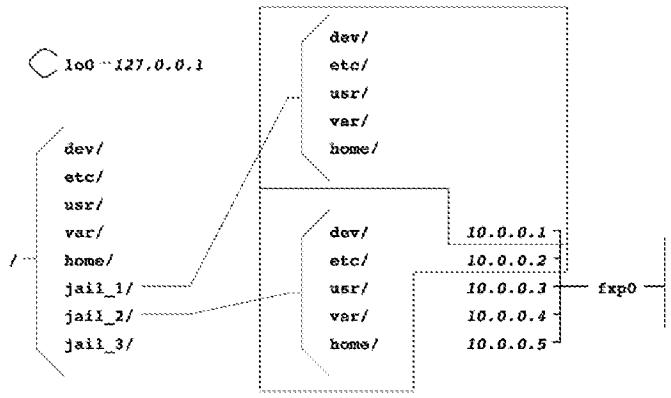


Fig. 1 — Schematic diagram of machine with two configured jails

Kamp (EX1026), 5. Each jail in a FreeBSD operating system operates in a limited scope, unable to access other jails or the root system: "Jail neatly side-steps the majority of these problems through partitioning. Rather than introduce additional fine-grained access control mechanism, we partition a FreeBSD environment (processes, file system, network resources) into a management environment, and optionally subset Jail environments. In doing so, we simultaneously maintain the existing UNIX security model, allowing multiple users and a privileged root user in each jail, while **limiting the scope of root's activities to his jail**." Kamp (EX1026), 4. EX1003, ¶ 85.

In essence, processes operating in one jail cannot use processes in a different jail (i.e., processes operating in jail_1 utilize resources such as libraries, processes, file system, and network resources (*an instance of a SLCSE*) which are not *shared with other of the plurality of software applications* operating in other jails, such as jail_2). When in a jail, applications are not able to

leave the jail or access resources available outside of the jail in which the application is run: "A process in a partition is referred to as 'in jail'. When a FreeBSD system is booted up after a fresh install, no processes will be in jail. When a process is placed in a jail, it, and any descendents of the process created after the jail creation, will be in that jail. **A process may be in only one jail, and after creation, it can not leave the jail**." Kamp (EX1026), 4. "Membership in a jail involves a number of restrictions: access to the file name-space is restricted in the style of chroot(2), the ability to bind network resources is limited to a specific IP address, **the ability to manipulate system resources and perform privileged operations is sharply curtailed, and the ability to interact with other processes is limited to only processes inside the same jail**." Kamp (EX1026), 4. "**Processes within the jail will find that they are unable to interact or even verify the existence of processes outside the jail** — processes within the jail are prevented from delivering signals to processes outside the jail, as well as connecting to those processes with debuggers, or even see them in the sysctl or process file system monitoring mechanisms." Kamp (EX1026), 5. EX1003, ¶ 86.

Accordingly, in the combination, a system implementing NetTap, per Blott, would have instances of the libraries (*SLCSEs*) in each of jail_1 and jail_2 created, per Kamp. Each of jail_1 and jail_2 run separate instances of software applications that access separate instances of the NetTap libraries, as the instances of the libraries can only be accessed by programs running within their respective jails. EX1003, ¶ 87.

Thus, Lucas in view of Blott and Kamp renders obvious *wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library* (a copy of the libraries, processes, file system, and network resources within FreeBSD, including NetTap, are provided to FreeBSD jail_1) *is run in a context of said at least first of the plurality of software applications* (applications operating in a FreeBSD jail_1 use jail_1's libraries, processes, file system, and network resources, including NetTap) *without being shared with other of the plurality of software applications* (i.e., software operating outside of FreeBSD jail_1 cannot access the libraries, processes, file system, and network resources inside jail_1, including NetTap) *and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function* (FreeBSD jail_2 can be run with a unique copy of the libraries, processes, file system, and network resources, including NetTap, used by applications running in jail_2). EX1003, ¶¶ 77-88.

[1.6] iii) wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously.

Lucas in view of Blott and Kamp renders obvious *wherein a SLCSE related to a predetermined function* (Blott's NetTap in jail_1) *is provided to the first of the plurality of software applications for running a first instance of the SLCSE* (jail_1 has a copy of the libraries, processes, file system, and network resources, including NetTap, which are used by software operating within jail_1)*, and wherein a SLCSE for performing a same function* (a second instance of NetTap in jail_2) *is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously* (jail_2 has a copy of the libraries, processes, file system, and network resources, including NetTap, which are used by software operating within jail_2 at the same time as software operating within jail_1). EX1003, ¶¶ 89-93.

As discussed above in the analysis of element [1.5], in the combination, jail_1 and jail_2, as described by Kamp, each contain copies of the libraries, processes, file system, and network resources, which are provided to software applications running within the respective jails without being shared with applications running in other jails or the root system:

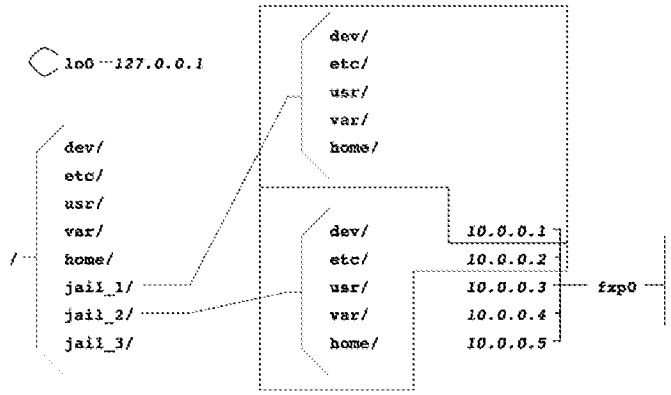


Fig. 1 — Schematic diagram of machine with two configured jails

Kamp (EX1026), 5. EX1003, ¶ 90.

Accordingly, a copy of the libraries, processes, file system, and network resources within FreeBSD, including NetTap, are provided to FreeBSD jail_1. The NetTap libraries are *SLCSE[s] related to a predetermined function*, as the libraries provide critical system elements related to IP forwarding and bridging. *See* Blott (EX1008), 1, 3, 8. These libraries are provided to software

applications running within jail_1, and jail_1 contains instances of the libraries which are not shared with software outside of the jail (i.e., the libraries are *provided to the first of the plurality of software applications for running a first instance of the SLCSE*) because applications are limited in scope to operating and accessing resources within jail_1. *See* Kamp (EX1026), 4 ("limiting the scope of root's activities to his jail"); *see also* Kamp (EX1026), 11 (explaining that "[a]s with any FreeBSD system, accounts may be created and deleted, mail is delivered, logs are generated, packages may be added, and the system may be hacked into if con- figured incorrectly, or running a buggy version of a piece of software. However, **all of this happens strictly within the scope of the jail**."). EX1003, ¶ 91.

A second copy of the libraries, processes, file system, and network resources within FreeBSD, including NetTap, are provided to FreeBSD jail_2. These libraries are provided to software applications running within jail_2, and jail_2 contains instances of the libraries which are not shared with software outside of the jail (i.e., the libraries are *provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously*). Indeed, it is expected that software applications are running on multiple jails simultaneously. *See* Lucas (EX1004), 181 ("To investigate **individual process to learn which jail they're part of**, use the process filesystem, procfs."). In the combined system implemented Blott's NetTap, which provides advantageous performance over kernel level IP forwarding and bridging (*See* Blott (EX1008), 1, 8-9), it would be obvious that programs running simultaneously on jail_1 and jail_2 would be accessing respective instances of NetTap provided to each jail. EX1003, ¶ 92.

Thus, Lucas in view of Blott and Kamp renders obvious *wherein a SLCSE related to a predetermined function* (Blott's NetTap in jail_1) *is provided to the first of the plurality of software applications for running a first instance of the SLCSE* (jail_1 has a copy of the libraries, processes, file system, and network resources, including NetTap, which are used by software operating within jail_1)*, and wherein a SLCSE for performing a same function* (a second instance of NetTap in jail_2) *is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously* (jail_2 has a copy of the libraries, processes, file system, and network resources, including NetTap, which are used by software operating within jail_2 at the same time as software operating within jail_1). EX1003, ¶¶ 89-93.

2. Claim 12

[12.0] A computing system according to claim 1, wherein SLCSEs include services related to at least one of, network protocol processes, and the management of files.

Lucas in view of Blott and Kamp renders obvious *a computing system according to claim 1,* (Lucas, Blott, and Kamp render obvious claim 1) *wherein SLCSEs include services related to at least one of, network protocol processes, and the management of files* (Blott's NetTap provides user-level bridging and IP forwarding through user-level libraries for TCP/IP protocols). EX1003, ¶¶ 94-96.

First, as discussed above in the analysis of elements [1.0]-[1.6], the combination of Lucas, Blott, and Kamp renders obvious *a computing system according to claim 1.* Second, as discussed in the analysis of element [1.4], above, Blott "provides user-level libraries for TCP/IP protocols, IP security [16], and output link scheduling [7]," which it refers to as NetTap (e.g., *SLCSEs*). Blott (EX1008), 6. Blott's NetTap provides *functional replicas of OSCSEs* such as FreeBSD kernel level functions for IP forwarding at the user level. NetAPP (*SLCSEs*) *include[s] services related to...network protocol processes* such as TCP/IP protocols, and more specifically, provides user-level bridging and IP forwarding replicas of the kernel level bridging and IP forwarding in FreeBSD. *See* Blott (EX1008), 1, 3, 8. EX1003, ¶ 95.

Thus, Lucas in view of Blott and Kamp renders obvious *a computing system according to claim 1,* (Lucas, Blott, and Kamp render obvious claim 1) *wherein SLCSEs include services related to at least one of, network protocol processes, and the management of files* (Blott's NetTap provides user-level bridging and IP forwarding through user-level libraries for TCP/IP protocols). EX1003, ¶¶ 94-96.

B. Secondary Considerations

This Request demonstrates that the Challenged Claims of the '058 Patent are unpatentable as obvious in view of the prior art references. The Applicant did not identify any evidence of secondary considerations during prosecution. Further, the clear teachings in the prior art outweigh any supposed "secondary considerations." *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 36 (1966).

III. DISCLOSURE OF CONCURRENT LITIGATION, REEXAMINATION, AND RELATED PROCEEDINGS

Based on information available to Requester, the '058 Patent is the subject of 5 District Court litigations, as listed in the below table. Requester is unaware of any prior reexaminations or other post-grant proceedings in which the '058 Patent is or has been involved.

- *VirtaMove, Corp. v. Hewlett Packard Enterprise Company*, Case No. 2:24-cv-00093 (E.D. Tex., Feb. 9, 2024)
- *VirtaMove, Corp. v. Amazon.com, Inc. et al*, Case No. 7:24-cv-00030 (W.D. Tex., Jan. 26, 2024)
- *RedHat, Inc. v. VirtaMove, Corp.*, Case No. 5:24-cv-04740 (N.D. Cal., Aug. 5, 2024)
- *VirtaMove, Corp. v. Google LLC*, Case No. 7:24-cv-00033 (W.D. Tex., Jan. 31, 2024)
- *VirtaMove, Corp. v. International Business Machines, Corp.*, Case No. 2:24-cv-00064 (E.D. Tex., Jan. 31, 2024)

IV. CONCLUSION

The Commissioner is hereby authorized to charge Deposit Account 50-6990 under Docket No. VIR058 the *Ex Parte* Reexamination fee of $12,600 under 37 C.F.R. § 1.20(c)(2). Requester believes no other fee is due with this submission, however the Commissioner is hereby authorized to charge any fee deficiency or credit any over-payment to Deposit Account 50-6990.

Please direct all correspondence in this matter to the undersigned.

Dated: 9/23/24

Respectfully submitted,

By: /Michelle Aspen/
Michelle Aspen, Reg. No. 75,665
Matthew Beck, Reg. No. 79,791
Roshan Mansinghani, Reg. No. 62,428
4445 Willard Avenue, Suite 600
Chevy Chase, MD 20815
michelle@unifiedpatents.com
T: 559-214-3388
matt@unifiedpatents.com
T: 240-223-7995
roshan@unifiedpatents.com
T: 214-953-6737

Counsel for Requester Unified Patents, LLC