

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION
Petitioner,

v.

EDGE NETWORKING SYSTEMS, LLC,
Patent Owner.

U.S. Patent No. 10,893,095
Issue Date: January 12, 2021
Title: DISTRIBUTED SOFTWARE DEFINED NETWORKING

Inter Partes Review No.: IPR2025-00617

**PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO.
10,893,095 UNDER 35 U.S.C. §§ 311-319 and 37 C.F.R. §§ 42.1-.80, 42.100-
.107**

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

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. OVERVIEW	1
III. GROUNDS FOR STANDING (37 C.F.R. §42.104(A))	6
IV. RELIEF REQUESTED (37 C.F.R. §42.22(a)).....	6
V. REASONS FOR THE REQUESTED RELIEF	6
A. '095 Patent Summary	6
B. Prosecution History	10
C. Claim Construction	10
D. Priority Date of the Challenged Claims	13
E. Person of Ordinary Skill in the Art	14
F. State of the Art	14
1. Sandboxing Operating Systems	14
2. Security for Software Downloaded/Implemented over the Internet	16
3. Distributed Systems	17
VI. IDENTIFICATION OF CHALLENGES.....	23
A. Challenged Claims	23
B. Statutory Grounds for Challenges	23
VII. IDENTIFICATION OF HOW THE CHALLENGED CLAIMS ARE UNPATENTABLE	25
A. Ground 1: Claims 1-2, 6-8, 10-13 and 15 are obvious over Vasell in view of Alves, Hall and Rellermeyer.....	25

1.	Vasell.....	25
2.	Alves	31
3.	Hall.....	33
4.	Rellermeyer.....	34
5.	The Motivation to Combine.....	36
6.	Detailed Application of Vasell, Alves, Hall and Rellermeyer to the Challenged Claims	42
	Claim 1	42
	Claim 2.....	64
	Claim 6.....	65
	Claim 7.....	66
	Claim 8.....	67
	Claim 10.....	70
	Claim 11	71
	Claim 12.....	72
	Claim 13.....	72
	Claim 15.....	74
VIII.	Discretionary Denial Would be Inappropriate	79
	A. Discretionary Denial under the <i>Fintiv</i> Factors is Inappropriate.....	79
	B. Discretionary Denial Under §325(d) is Not Appropriate.....	82
IX.	Mandatory Notices Under 37 C.F.R. §42.8.....	82
	A. Real Party-in-Interest (37 C.F.R. § 42.8(b)(1)).....	82
	B. Related Matters (37 C.F.R. § 42.8(b)(2)).....	82

1.	Judicial Matters	82
2.	Related Patents	83
C.	Lead/Back-up Counsel (37 C.F.R. § 42.8(b)(3)):	83
D.	Notice of Service Information (37 C.F.R. § 42.8(b)(4)):	84
X.	CONCLUSION	84
	CERTIFICATE OF COMPLIANCE	1
	CERTIFICATION OF SERVICE ON PATENT OWNER	2

PETITIONER’S EXHIBIT LIST

<i>Exhibit #</i>	<i>Description</i>
1001	U.S. Patent No. 10,893,095 (the “’095 Patent”).
1002	Prosecution history of the ’095 Patent (the “’095 File History”).
1003	Declaration of Petitioner’s Expert, Dr. Erez Zadok, (“Zadok”).
1004	U.S. Patent No. 6,496,575 to Vasell, et al. (“Vasell Patent”) (collectively, with EX1015 and EX1016, “Vasell”).
1005	File history of the parent patent to the ’095 Patent, U.S. Patent No. 9,843,624 (the “’624 Patent”).
1006	File history of another parent patent of the ’095 Patent, U.S. Patent No. 10,686,871 (the “’871 Patent”).
1007	File history of a continuation patent of the ’095 Patent, U.S. Patent No. 11,695,823 (the “’823 Patent”).
1008	Excerpts from A. de Castro Alves, “OSGi in Depth”, Manning Publications Company, 2012 (“Alves”).
1009	Excerpts from R. Hall, et al., “OSGi in Action - Creating Modular Applications in Java”, Manning Publications Company, April 2011 (“Hall”).
1010	J. E. Kim, et al., “Seamless Integration of Heterogeneous Devices and Access Control in Smart Homes,” 2012 Eighth International Conference on Intelligent Environments, Guanajuato, Mexico, June 26-29, 2012, pp. 206-213 (“Kim”).
1011	J. S. Rellermeyer and S. Bagchi, “Dependability as a cloud service - a modular approach,” IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012), Boston, MA, USA, 2012, pp. 1-6 (“Rellermeyer”).
1012	S. Kächele, et al., “Component-based scalability for cloud applications”, Proceedings of the 3 rd International Workshop on Cloud Data and Platforms (CloudDP ‘13), April 14, 2013, Prague, Czech Republic, 19–24 (“Kachele”).
1013	OSGi Alliance, “About the OSGi Service Platform”, Technical Whitepaper, Revision 4.1, November 11, 2005, archived on November 30, 2005 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20051130032628/http://www.osgi.org/documents/collateral/TechnicalWhitePaper2005osgi-sp-overview.pdf (“OSGi 2005 Whitepaper”).

<i>Exhibit #</i>	<i>Description</i>
1014	Declaration of June A. Munford.
1015	U.S. Provisional Patent Application No. 60/088,437, filed on June 8, 1998, claimed as a priority application in, and incorporated by reference into, EX1004 (“1 st Vasell Provisional”) (collectively, with EX1004 and EX1016, “Vasell”).
1016	U.S. Provisional Patent Application No. 60/123,971, filed on March 12, 1999, claimed as a priority application in, and incorporated by reference into, EX1004 (“2 nd Vasell Provisional”) (collectively, with EX1004 and EX1015, “Vasell”), which includes a copy of the article published as Idermark, T., Lilliestråle, M., & Vasell, J., “Ericsson's e-box system—An electronic services enabler”, Ericsson Review, (1), 1999, 38-44, as confirmed by Ericsson’s website archived, for example, on March 3, 2000 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20000303195310/http://www.ericsson.se/review/issues.taf and, for example, on February 16, 2003 and November 27, 2004 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20030216191220/http://www.ericsson.com/about/publications/review/1999_01/23.shtml and https://web.archive.org/web/20041127035720/http://www.ericsson.com/about/publications/review/1999_01/files/1999015.pdf respectively.
1017	“About Gatespace”, Gatespace AB website, archived on January 6, 2000 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20000106145021/http://www.gatespace.com/ (“About Gatespace 2000”).
1018	“Carlstedt Research & Technology Forms New Company for Developing Software for ‘The Intelligent Home’ in Cooperation with Ericsson”, September 30, 1999, Gatespace AB website, archived on February 12, 2001 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20010212054449/http://www.gatespace.com/news/archive/19990930_en_v3.shtml (“Gatespace Formation Press Release”).

<i>Exhibit #</i>	<i>Description</i>
1019	“Ericsson e-services is a part of the Open Services Gateway Initiative (OSGI)”, Ericsson e-services website, archived on February 29, 2000 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20000229111550/http://www.ericsson.com:80/wireless/products/ebox/links/osgi.shtml (“Ericsson E-Services WebPage”).
1020	“Fifteen industry leaders to create standard for bringing Internet-based services to the networked home”, March 1, 1999, Ericsson e-services website, archived on May 28, 2000 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20000528223608/http://www.ericsson.com/wireless/products/ebox/news/990301.shtml (“OSGi Announcement on Ericsson E-Services WebPage”).
1021	“People”, Carlstedt Research & Technology AB website, archived on January 6, 2000 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20000106204945/http://www.crt.se/about/people.en.html (“CR&T People 2000”).
1022	“Open Services Gateway Initiative: Charter”, OSGi website, archived on April 19, 2000 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20000419181219/http://www.osgi.org/about/charter.html (“OSGi Charter 2000”).
1023	“Open Services Gateway Initiative: Officers”, OSGi website, archived on April 14, 2000 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20000414133426/http://www.osgi.org/about/officers.html (“OSGi Officers 2000”).
1024	“Open Services Gateway Initiative Elects New Leadership”, Home Toys Inc. website, May 23, 2001, archived on July 8, 2001 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20010708123711/http://www.hometoys.com/releases/jun01/osgi01.htm (“OSGi Leadership 2001”).
1025	“Management Team”, Gatespace AB website, archived on February 8, 2002 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20020208031748/http://www.gatespace.com/company/management.shtml (“Gatespace Management 2002”).

<i>Exhibit #</i>	<i>Description</i>
1026	“About Us: Background”, Gatespace AB website, archived on February 6, 2002 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20020206205742/http://www.gatespace.com/company/ (“About Gatespace 2002”).
1027	“About Makewave: History in Making”, Makewave AB website, archived on August 16, 2007 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20070816163615/http://www.makewave.com/site.en/about/history.shtml (“About Makewave 2007”).
1028	The OSGi Alliance, “OSGi Service Platform Enterprise Specification”, Release 4, Version 4.2, March 2010 (“OSGi Enterprise Specification”).
1029	T. Forst, “Cisco Application eXtension Platform – OSGi Add-on as universal Middleware for Your Applications”, May 2008, AutomatedBuildings.com, archived on May 17, 2008 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20080517111602/www.automatedbuildings.com/news/may08/articles/prosyst/080427034829prosyst.htm (“Forst”).
1030	U.S. Patent Publication No. 2013/0086147 to Kashyap, filed October 3, 2011, published April 4, 2013 (“Kashyap”).
1031	Excerpts from A.S. Tanenbaum, “Modern Operating Systems”, Prentice Hall, 3 rd edition, 2008 (“Tanenbaum 2008”).
1032	Excerpts from A.S. Tanenbaum, “Computer Networks”, Prentice Hall, 3 rd edition, 1996 (“Tanenbaum 1996”).
1033	Excerpts from D. Downing, et al., “Dictionary of Computer and Internet Terms”, Barron’s Educational Series, Inc., 11 th edition, 2013 (“Downing”).
1034	Excerpts from E. Cole, et al., “Network Security Bible”, Wiley Publishing, Inc., 2 nd edition, 2009 (“Cole”).
1035	My Curriculum Vitae (CV).
1036	S. Soltesz, et al., “Container-based Operating System Virtualization: A Scalable, High Performance Alternative to Hypervisors”, In Proceedings of the 2 nd ACM SIGOPS/EuroSys European Conference on Computer Systems (EuroSys ’07), March 21-23, 2007, Lisbon, Portugal, pp. 275-287, 2007 (“Soltesz”).
1037	<i>Intentionally Left Blank</i>

<i>Exhibit #</i>	<i>Description</i>
1038	<i>Intentionally Left Blank</i>
1039	Li, et al., “ExpoNet: A Flexible Platform for Concurrent Experiments on Programmable Infrastructure”, 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011, Houston, TX, USA, 2011, pp. 1-5 (“Li”).
1040	<i>Intentionally Left Blank</i>
1041	P. Kriens, “The Bundle Repository”, OSGi Alliance Blog, April 7, 2006, archived on May 4, 2006 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20060504203641/http://www.osgi.org/blog/2006/04/bundle-repository.html (“Kriens OSGi Blog”).
1042	“Knoplerfish Pro Premium” and “Knoplerfish Pro Enterprise” Data Sheets, Makewave AB, 2010, respectively archived on February 20, 2011 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20110220123804/http://www.makewave.com:80/resources/docs/datasheets/MD-042-A1-knoplerfish_pro_premium_osgi.pdf and https://web.archive.org/web/20110220123819/http://www.makewave.com:80/resources/docs/datasheets/MD-047-A1-knoplerfish_pro_enterprise_osgi.pdf (“Knoplerfish”).
1043	R. Kawashima, “vNFC: A Virtual Networking Function Container for SDN-enabled Virtual Networks”, 2012 Second Symposium on Network Cloud Computing and Applications, London, UK, 2012, pp. 124-129 (“Kawashima”).
1044	Excerpts from A.S. Tanenbaum, “Modern Operating Systems”, Prentice Hall, 2 nd edition, 2001 (“Tanenbaum 2001”).
1045	Excerpts from S. Garfinkel, et al., “Web Security & Commerce”, O’Reilly & Associates, 1997 (“Garfinkel”).

<i>Exhibit #</i>	<i>Description</i>
1046	Oasis Standard Specification, “Web Services Security: SOAP Message Security 1.1 (WS-Security 2004), February 1, 2006, archived on February 6, 2007 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20070206143722/http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-soapmessagesecurity.pdf, and accessible from OASIS Web Services Security (WSS) Technical Committee web page, as archived on December 5, 2006 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20061205032600/http://oasis-open.org/committees/tc_home.php?wg_abbrev=wss#technical (“Oasis - SOAP Security”).
1047	“Web Service Security Cheat Sheet”, The Open Web Application Security Project (OWASP), archived on June 12, 2012 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20120612205800/https://www.owasp.org/index.php/Web_Service_Security_Cheat_Sheet, and accessible from OWASP’s (the free and open software security community) web pages, as archived on June 10, 2012 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20120610001928/https://www.owasp.org/index.php/Main_Page, and as archived on June 10, 2012 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20120610091415/https://www.owasp.org/index.php/Cheat_Sheets (“OWASP - WSS Security”).
1048	“REST Security Cheat Sheet”, The Open Web Application Security Project (OWASP), archived on January 15, 2013 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20130115190935/https://www.owasp.org/index.php/REST_Security_Cheat_Sheet, and accessible from OWASP’s (the free and open software security community) web pages, as archived on January 4, 2013 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20130104230717/https://www.owasp.org/index.php/Main_Page, and as archived on January 15, 2013 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20130115215253/https://www.owasp.org/index.php/Cheat_Sheets (“OWASP – REST Security”).

<i>Exhibit #</i>	<i>Description</i>
1049	<i>Intentionally Left Blank</i>
1050	The Open Services Gateway Initiative, “Specification Overview”, Version 1.0, January 2000, archived on August 31, 2000 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20000831061202/http://www.osgi.org/about/specoverview.pdf (“OSGi 2000 Overview”).
1051	R. Hall, “Oscar Bundle Repository”, archived on June 30, 2004 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20040630163519/http://oscar-osgi.sourceforge.net:80/ (“OBR 2004”).
1052	Excerpts from B. Sosinsky, “Cloud Computing Bible”, Wiley Publishing, Inc., 2011 (“Sosinsky”).
1053	“Java Secure Socket Extension (JSSE) Reference Guide for Java Platform Standard Edition 7”, Oracle Java SE Documentation, archived on November 25, 2011 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20111125044243/http://docs.oracle.com/javase/7/docs/technotes/guides/security/jsse/JSSERefGuide.html (“JSSE Guide”).
1054	“Java™ Remote Method Invocation API (Java RMI)”, Oracle Java SE Documentation, archived on November 27, 2011 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20111127072151/http://docs.oracle.com/javase/7/docs/technotes/guides/rmi/index.html (“Java RMI Overview”).
1055	“Using Custom Socket Factories with Java RMI”, Oracle Java SE Documentation, archived on April 3, 2012 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20120403095031/https://docs.oracle.com/javase/7/docs/technotes/guides/rmi/socketfactory/index.html (“Java RMI Socket Overview”).
1056	“Using Java™ RMI with SSL”, Oracle Java SE Documentation, archived on July 3, 2012 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20120703221253/https://docs.oracle.com/javase/7/docs/technotes/guides/rmi/socketfactory/SSLInfo.html (“Java RMI SSL Overview”).
1057	Microsoft Computer Dictionary, 3rd ed., 1997, excerpts (“Microsoft Computer Dictionary”).

<i>Exhibit #</i>	<i>Description</i>
1058	Usenix Jails in FreeBSD for Fun and Profit, Paco Hope, USENIX ;login: The Magazine of Usenix & Sage, vol. 27, number 3, June 2002, archived on March 16, 2003 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20030316225604/http://www.usenix.org/publications/login/2002-06/pdfs/hope.pdf (“Hope”).
1059	Excerpts from Silberschatz et al., Operating System Concepts (8 th Ed. 2009) (“Silberschatz 2009”).
1060	Excerpts from Silberschatz et al., Operating System Concepts (9 th Ed. 2012) (“Silberschatz 2012”).
1061	Excerpts from McKusick and Neville-Neil, “The Design and Implementation of the FreeBSD Operating System” (2005) (“McKusick”).
1062	Excerpts from Benevenuti, “Understanding Linux Network Internals” (2006) (“Benevenuti”).
1063	Excerpts from Stallings, “Operating Systems Internals and Design Principles” (7 th ed. 2012) (“Stallings”).
1064	“apache_1.3.19.tar.gz.md5” Hash File, Apache Software Foundation, archived on June 28, 2001 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20010628043950/http://www.apache.org/dist/httpd/apache_1.3.19.tar.gz.md5 .
1065	“apache_1.3.19.tar.gz.asc” Signature File, Apache Software Foundation, archived on June 28, 2001 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20010628043146/apache.org/dist/httpd/apache_1.3.19.tar.gz.asc .
1066	“Apache Archive Distribution Directory”, Apache Software Foundation, archived on July 11, 2001 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20010711223314/http://www.apache.org/dist/httpd/ (“Apache Archive Directory”).
1067	U.S. Department of Commerce, FIPS PUB 186-2, Digital Signature Standard (DSS) (January 27, 2000) (“DSS FIPS PUB”).

<i>Exhibit #</i>	<i>Description</i>
1068	J. Preshing, “Hash Collision Probabilities”, Preshing on Programming, May 4, 2011, archived on September 27, 2011 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20110927200414/preshing.com/20110504/hash-collision-probabilities (“Preshing Hash Collision Probabilities”).
1069	“Avalanche Effect”, Wikipedia, archived on April 30, 2010 at Internet Archive’s Wayback Machine at https://web.archive.org/web/20100430122132/https://en.wikipedia.org/wiki/Avalanche_effect (“Wikipedia Avalanche Effect”).
1070	Plaintiff Edge Networking Systems, LLC’s Opening Claim Construction Brief, filed as Document No. 31, on December 17, 2024, in Case No. 1:24-cv-00215-DAE, captioned as <i>Edge Networking Systems, LLC v. Microsoft Corporation</i> (W.D. Texas).
1071	August 12, 2024 USPTO Denial of Applicant’s December 1, 2022 Request for Certificate of Correction of the ’095 Patent (“’095 Patent File History Addendum”).
1072	Defendant Microsoft Corporation’s Responsive Claim Construction Brief, filed as Document No. 32, on January 17, 2025, in Case No. 1:24-cv-00215-DAE, captioned as <i>Edge Networking Systems, LLC v. Microsoft Corporation</i> (W.D. Texas).
1073	Scheduling Order, filed as Document No. 27, on September 4, 2024, in Case No. 1:24-cv-00215-DAE, captioned as <i>Edge Networking Systems, LLC v. Microsoft Corporation</i> (W.D. Texas).

I. INTRODUCTION

Pursuant to 35 U.S.C. §§311 *et seq.* and 37 C.F.R. §§42.1 *et seq.*, Microsoft Corporation (“Petitioner”) petitions for an *inter partes* review (“IPR”) of U.S. Patent No. 10,893,095 (“’095 Patent”). Petitioner submits that ’095 Patent Claims 1-2, 6-8, 10-13, and 15 (the “Challenged Claims”) are unpatentable under 35 U.S.C. §103 in view of the prior art references discussed herein. This Petition demonstrates by a preponderance of the evidence that there is a reasonable likelihood that Petitioner will prevail for at least one of these claims. Accordingly, Petitioner requests that the Board institute an IPR of the ’095 Patent pursuant to 37 C.F.R. §42.108.

II. OVERVIEW

Throughout this Petition, the following color scheme is used to annotate the same elements in the ’095 Patent and the prior art¹:

Concept	Color
distributed system for processing data packets comprising a network device and a cloud device	alternating green and maroon
network device	green
cloud device	maroon

¹ Petitioner has added all emphasis in quoted materials and annotations to the figures.

distributed application with a component on a network device and a component on a cloud device	alternating pink and orange
first network application that is a component of the distributed application hosted on a network device	pink
second network application that is a component of the distributed application hosted on a cloud device	orange
application management portal (and associated software)	blue
application repository/store	brown

The Challenged Claims recite “a system for processing data packets in a network.” The system includes programmable network and cloud devices spread across a network, a distributed software application with components installed on each of these devices, “upgrades” of these components with minimal interruption to these devices’ operation, an application repository “storing distributed applications”, and an “application management portal” “managing usage of the distributed applications.”

The '095 Patent calls a system with this functionality a “distributed software-defined network (dSDN).” . FIG. 3 illustrates one dSDN embodiment:

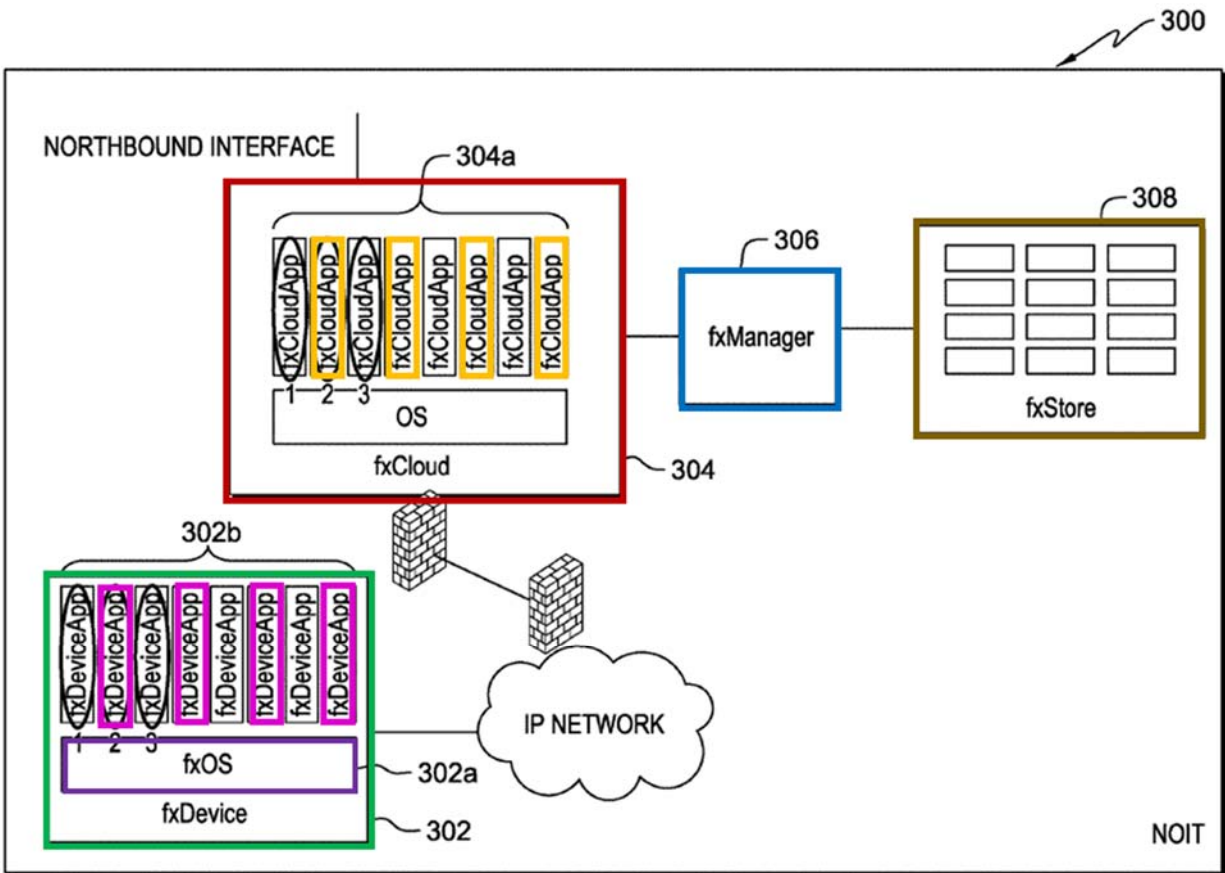


FIG. 3

Figure 3 shows “distributed applications” 1, 2, and 3 (among others), and the two components of distributed applications 1, 2, and 3 respectively installed on a “network device” and a “cloud device.” Figure 3 also shows an “IP Network” across which the “network device” and “cloud device” are distributed, an “application management portal” managing the distributed applications’ lifecycle and “manag[ing] upgrades” of the distributed applications’ components, and an “application repository” storing applications “for installation in” the “network device” and “cloud device.”

The Challenged Claims, when stripped of their jargon, claim a system including (i) various **devices** spread across a network, (ii) a distributed **software application** having components installed on these **devices**, and (iii) a **management interface** and **repository** that allows a user to download and upgrade the distributed **software applications**' components.

This basic design was well-known and widely implemented before June 2013. Vasell (EX1004/EX1015/EX1016), filed in 1998-1999, provides an example of such a **distributed system** and distributed **software applications**. Like the Patent, Vasell discloses a system for processing data packets having **devices** spread across the Internet, called a "**service gateway system**." Vasell discloses a "**network device**" called "**service platform server**," and a "**cloud device**," called "**service provider equipment**" or "**network operator server**." Vasell discloses a plurality of distributed **software applications**, each having a **component** installed on a **network device** and a **component** installed on a **cloud device**. And Vasell teaches a **management service** having an **external interface** to manage the distributed **applications**' lifecycle and manage upgrades of the distributed **applications**' components. Vasell's Annotated FIG. 2 shows one "**service gateway system**" embodiment:

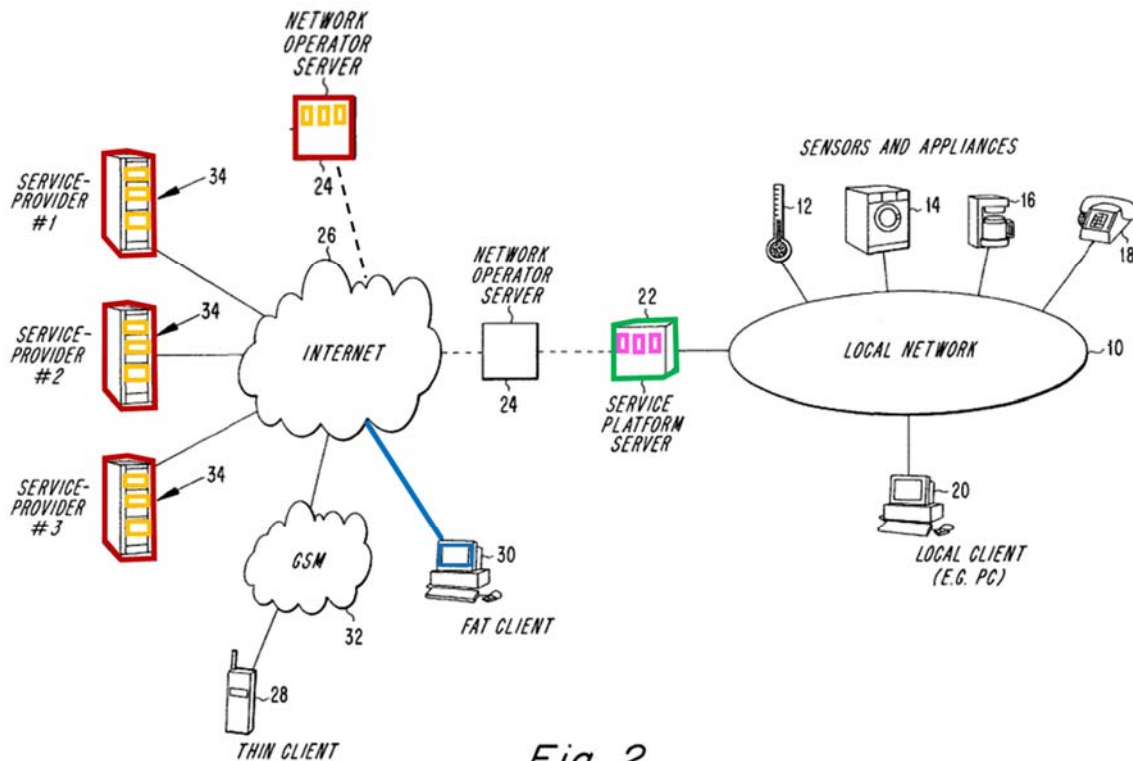


Fig. 2

Unsurprisingly, Vasell leaves certain implementation details to design choices of POSITAs. However, Vasell’s architecture and methods were foundational concepts (including implementation in a Java run-time environment) that were eventually standardized under the Open Standards Gateway initiative (OSGi) specifications. Technical details that were intended to implement Vasell’s teachings were described in each of the published references cited in Ground 1—Alves (EX1008), Hall (EX1009) and Rellermeyer (EX1011). Accordingly, POSITAs would naturally combine these references’ teachings together, and their combined teachings disclose each limitation of the Challenged Claims.

III. GROUNDS FOR STANDING (37 C.F.R. §42.104(A))

Petitioner certifies the '095 Patent is available for IPR and Petitioner is not barred or estopped from requesting an IPR of the Challenged Claims on the grounds identified herein. 37 C.F.R. §42.104(a). This Petition is filed pursuant to 37 C.F.R. §42.106(a).

IV. RELIEF REQUESTED (37 C.F.R. §42.22(a))

Petitioner respectfully requests institution of an IPR pursuant to 37 C.F.R. §42.108 and cancellation of the Challenged Claims.

V. REASONS FOR THE REQUESTED RELIEF

As explained in §II and §VI-§VIII of this Petition and in the attached Declaration of Petitioner's Expert, Dr. Erez Zadok, EX1003, the **distributed systems** implementing distributed **applications**, described and claimed in the '095 Patent, were obvious over the prior art to persons of ordinary skill in the art ("POSITAs") in June 2013.

A. '095 Patent Summary

The Challenged Claims are directed to a system for processing data packets that the Patent called a "**distributed software-defined network (dSDN)**." FIG. 3 (below) illustrates "a high-level overview of a **dSDN** system 300":

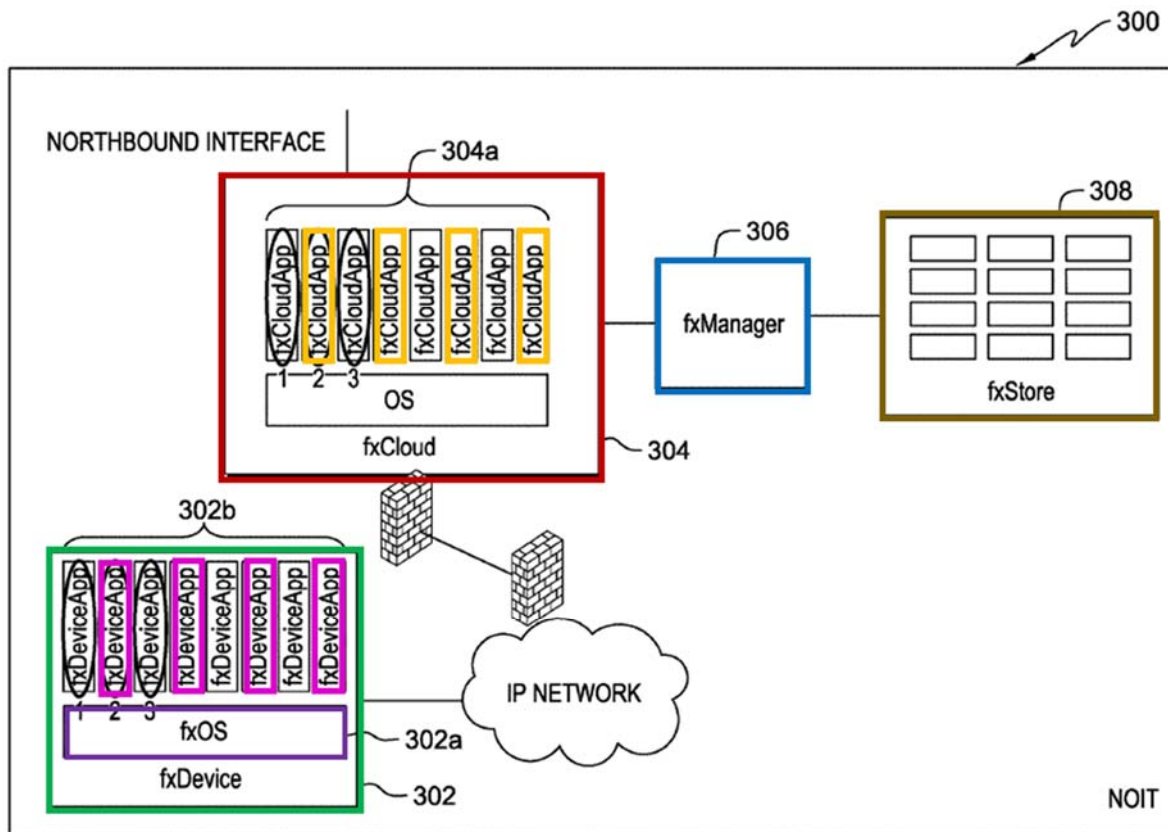


FIG. 3

EX1001, 10:8-13.

DSDN system 300 includes two hardware devices: “flexible network device (fxDevice)” and “flexible cloud platform (fxCloud).” The fxDevice is “any networking equipment... powered by a sandboxing operating system named flexible operating system (fxOS).” EX1001, 10:13-15. The fxCloud is “backend cloud infrastructure” (e.g., “network servers”), which could be a private, or “shared”/public, cloud platform. EX1001, 10:26-27, 11:32-33, 24:11, 25:45-48, 3:17-19, 17:67-18:3, 24:63-25:2.

Beyond the **system**'s hardware components, the Patent also described the distributed **applications** (e.g., 1, 2, 3 (FIG. 3, above)) that run across these **devices**. Each distributed **application** is separated into a **component** installed on an **fxDevice** (called "**fxDeviceApp**") and a component installed on an **fxCloud** device (called "**fxCloudApp**"). EX1001, 10:21-31, 18:52-53, 19:13-33.

In other words, a distributed **application** includes an **fxCloudApp component** "paired with" an **fxDeviceApp component**. EX1001, 10:21-31; EX1003, ¶¶245-248.

The Challenged Claims require the "**network device**" and "**cloud device**" to be "powered by a sandboxing operating system" (e.g., fxOS). Given that the **system** has a plurality of distributed **applications** run across these **devices**, a sandboxing operating system ("OS") ensures that the respective components of each distributed **application** installed on a given device run in a "dedicated process in sure isolation" from the other components (installed on that device) of the other distributed **applications**. EX1001, 10:25-46. Given this component-isolation feature of the sandboxing OS, the Patent describes the sandboxing OS as also "allow[ing] hot upgrade of the software and applications on the platform with little to no interruption to the operational aspect of the platform and its applications." EX1001, 11:29-32, 19:36-20:44, FIG. 13; EX1003, ¶¶254-255.

The Patent describes the “Virtual Fabric (fxVF)” dSDN “service” as “provid[ing] an abstraction layer for applications to communicate with each other whether they are in the fxDevice or fxCloud”, and “enabl[ing] transparent switching of application and system messages of a dApp between the fxCloud 304 and the fxDevice 302 and between different dApps”:

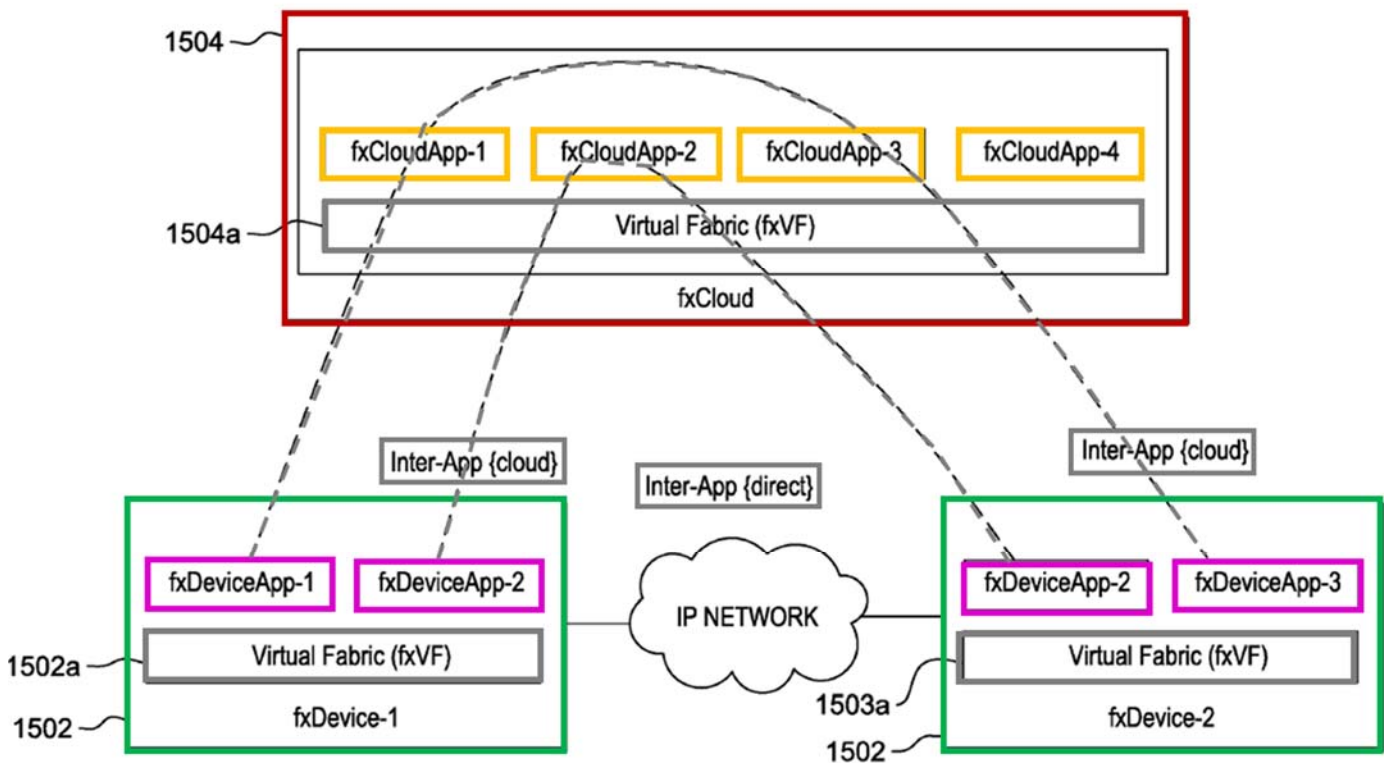


FIG. 15

EX1001, 20:48-59, 13:35-38, FIG. 15 (above), 10:44-50; EX1003, ¶¶256-259.

dSDN system 300 includes “application management portal (fxManager)”, which “presents a user-friendly portal to... network administrator[s]”, through which administrators may, for example, manage the fxDevice(s) and fxCloud server(s), perform “[a]pplication lifecycle management” tasks, and/or “brows[e]

through the certified applications in the **fxStore**.” EX1001, 15:13-16:34, 10:47-51, 17:58-19:34.

The Patent describes that, using well-known digital signature techniques, an application could be “signed” (so that, if a signature match is obtained by the recipient (*e.g.*, **fxManager**) using these same techniques, the application is certified to be authentic and intact (unchanged)). EX1001, 23:19-24:5; EX1003, ¶263.

dSDN system 300 includes “**fxStore**”, which may be “third party hosted”, contains software applications, and associated updates, developed (and optionally tested) for use in the **dSDN** environment by application developers, and from which a network administrator may download such applications/updates (*e.g.*, via **fxManager**). EX1001, 10:12-13, 10:51-55, 13:2-4, 15:43-44, 15:52-53, 16:32-41.

B. Prosecution History

The Challenged Claims received only a double patenting rejection, which was overcome by a terminal disclaimer. EX1002, 115-116; 120, 128.

C. Claim Construction

Petitioner proposes that each claim term in the Challenged Claims be given its plain and ordinary meaning in this proceeding, and no specific construction of any claim term is required because the prior art relied on in this Petition meets each of the claim terms under any reasonable construction. EX1003, ¶360. The parties have identified and briefed different constructions for various claim terms pending

in the Related Litigation. *See* EX1070; EX1072. However, the Petition Grounds render the claims unpatentable under either party’s construction, and any reasonable construction; thus the Board need not construe any term.

Petitioner addresses the below terms of the Challenged Claims to the extent Patent Owner (“PO”) argues or the Board finds they are governed by means-plus-function. Solely for purposes of this Petition, Petitioner relies on PO’s identification (EX1070) of “specific portions of the specification that describe the structure” corresponding to the recited functions. *Id.*; EX1003, ¶¶361-376.

- “virtual fabric” (claim 6): “an abstraction layer for applications to communicate with each other” per EX1001, 13:35-41, 20:46-51, 11:57-62. EX1070, 9-10; EX1072, 12-13.
- “the programmable network device and programmable cloud device are each powered by a sandboxing operating system which facilitates deployment of the plurality of first and second network applications” ([1.4], [15.4]): “the programmable network device and programmable cloud device each having a sandboxing [OS] allowing applications to run and interact with resources on the respective device” per EX1001, 3:59, 3:23-24, 10:59-62, 10:13-17, 11:1-3, 11:21-26, 11:50-62. EX1070, 10-12; EX1072, 14-15.

- “[sandboxing operating system]... facilitates upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device” ([1.5], [15.5]): “[t]he sandboxing [OS] ensures each application runs as a dedicated process [to] [ensure] isolation from the other applications,’ which permits one instance of the application to remain running during the upgrade, thereby permitting an upgrade to be carried out without substantial interruption” disclosed at EX1001, 11:29-32 (“hot upgrade of the software and applications on the platform with little to no interruption **to the operational aspect of the platform and its applications**”, 19:36-20:44, including 19:38-20:3 (in which the described process result is “the software upgrades on the system have no or minimal implications on the functionalities **offered by the [running] software component subject to the upgrade process**”), and 20:5-44 in which the described process result is “ensure that **major functionalities of the [programmable network device/programmable cloud device]** remain intact during the upgrade process.” EX1070, 12-13; EX1072, 15-17.
- “programmable network device” and “programmable cloud device”

“verifies authenticity of the upgrades... based on unique security keys

associated with each of the plurality of first and second network applications” ([1.6], [15.6]): “applications are being verified with a signature generated with a security key”, per EX1001, 11:25-29, 23:20-24:5. EX1070, 14-15; EX1072, 17-18.

- “the application management portal... manages [...]usage[...] of the distributed applications...” ([1.8], [15.8]): “a general-purpose computer with Internet access” adapted to carry out an algorithm of “using information available within a secure manifest file and elsewhere to ensure the availability of sufficient resources” as described in EX1001, 18:60-19:2. EX1070, 15-16; EX1072, 18-19.
- “load controller adapted to monitor loads ... and affect change in accordance with thresholds received ...” (claim 13): “controller executing at least one of a ‘cloud breathing’ or ‘load monitoring’ application” per EX1001, 22:10-15, 22:31-33 (load monitoring), 16:3, 22:20-37, FIGS. 17A-17B (cloud breathing). EX1070, 16-19; EX1072, 19-20.

D. Priority Date of the Challenged Claims

Petitioner assumes the Patent’s earliest-claimed priority date—June 13, 2013—is the priority date.

E. Person of Ordinary Skill in the Art

POSITAs as of June 2013 would have had a bachelor's degree in computer science, computer engineering, or equivalent degree, and approximately three years of experience working in the computer science or engineering field.

Additional experience might substitute for less education and vice versa. To that end, POSITAs in June 2013 would have been knowledgeable about the design and management of networked systems and virtualization technologies, and familiar with operating/distributed systems and security and privacy techniques. EX1003, ¶59.

F. State of the Art

The following section describes the state of the art for distributed systems and virtualization technologies as of June 2013. The prior art references, and the discussions of what was known to POSITAs, provide the factual support for the general description of the state of the art at the time of the invention, and provide additional motivation to modify or combine the references. Accordingly, these references should be considered by the Board.

1. Sandboxing Operating Systems

The Challenged Claims recite a “sandboxing operating system”, which was well-known functionality to POSITAs.

Traditional computer OSs included mechanisms to “sandbox” software programs (*e.g.*, confining/restricting what resources an application accesses and

when). EX1059, 591; EX1058, 2; EX1003, ¶¶117-119. Additional “sandboxing” was well-known to be achieved through (i) virtualizing the entire OS, which allowed multiple instances of the same OS to be run simultaneously on the same hardware, or through (ii) virtual machine (“VM”) technology, which was more than 40 years old by June 2013, and allowed the same hardware to host multiple VMs, each potentially running a different OS. EX1058, 3; EX1031, 569-570; EX1057, 498; EX1061, 124-127; EX1052, 100-103; EX1003, ¶¶99, 112, 120-121, 126-129, 220.

VM technology provided the benefits of isolating programs running in different OSs, and permitted companies with many servers doing many different tasks to have fewer physical machines, which saved costs, facilitated software development, and allowed for load balancing across VMs on different servers. EX1031, 569-570; EX1057, 498; EX1003, ¶¶99, 112.

In 1991, Sun Microsystems created the “Java” software language, providing “a method to execute programs **without any platform dependence.**” EX1034, 304. Java’s security model provides a well-known example of a sandboxed OS, in that the runtime environment, typically a “Java Virtual Machine (JVM)”, resided on the host computer’s OS and confined all Java applications to an isolated area—the “sandbox.” Specifically, the JVM’s well-known “sandbox” provided a restricted environment for running untrusted code (restricting access to resources

inside the sandbox) alongside trusted code (full access to system resources).

EX1034, 304-305, 406; EX1045, 44-45; EX1044, 642-643; EX1009, 438;

EX1003, ¶¶131-136.

2. Security for Software Downloaded/Implemented over the Internet

The Challenged Claims recite “verif[ying] authenticity of... [application] upgrades... based on unique security keys associated with... [the] applications.”

This was a well-known security tool for software downloaded/implemented over the Internet.

One important security aspect of Java software and JVMs was using digital signatures. A “digitally signed” Java application was created and delivered using standardized techniques, and treated as if it was trusted code if the application recipient used the “public key”, available in the signer’s (*e.g.*, application developer’s) certificate, to successfully “decrypt” the signature (affirming that the matching private key, of the public-key-private-key-pair unique to the signer, originally signed the application). EX1045, 169-172; EX1057, 228, 432, 131, 145, 388-389, 122-123; EX1067; EX1003, ¶¶82-85, 88, 138-140.

Java’s use of “digital signatures” was widely implemented so that recipients/implementers of the application could verify the “authenticity” and “integrity” of the application (and updates), “when...downloaded” and “before...

install[ing] it.” *Id.*; EX1044, 641-644, 590-591; EX1045, 265-266; EX1050, 8-10; EX1009, 457; EX1003, ¶¶140-149.

3. Distributed Systems

The term “distributed system” is well-understood as “a software system built on top of a network”, where “the distinction between a network and a distributed system lies with the software (especially the [OS]), rather than the hardware”, and where the “user of a distributed system is not aware that there are multiple processors, it looks like a virtual uniprocessor.” EX1032, 2; EX1057, 153-154; EX1003, ¶¶150, 153-154.

a. Open Standards Gateway Initiative (OSGi)

A well-known initiative related to modular applications and distributed systems—the Open Standards Gateway Initiative (OSGi)—commenced with a three-year joint research project in the late-1990s between Ericsson and CR&T, from which, in 1999, a patent application describing foundational concepts (“Vasell”, EX1015/EX1016/EX1004, *see* §VII.A.1, §VII.A.5-§VII.A.6) was filed, and the non-profit OSGi alliance was founded. EX1015; EX1016, 3-9; EX1003, ¶¶179-190.

Over the succeeding years, technical details were developed and published in the OSGi’s open specifications (*e.g.*, EX1028) and numerous OSGi-related

textbooks (e.g., EX1009, EX1008), whitepapers (e.g., EX1013, EX1050) and articles (e.g., EX1029, EX1010, EX1012, EX1011). EX1003, ¶191.

Vasell depicted a **distributed system** “end-to-end architecture” using JVMs hosted on respective **devices**’ OSs, which was used as the foundational architecture for the first OSGi specification in January 2000. Illustrated below is FIG. 2 of Vasell (top) and FIG. 2 from the OSGi specification (bottom), each showing hardware **devices** spread across the Internet::

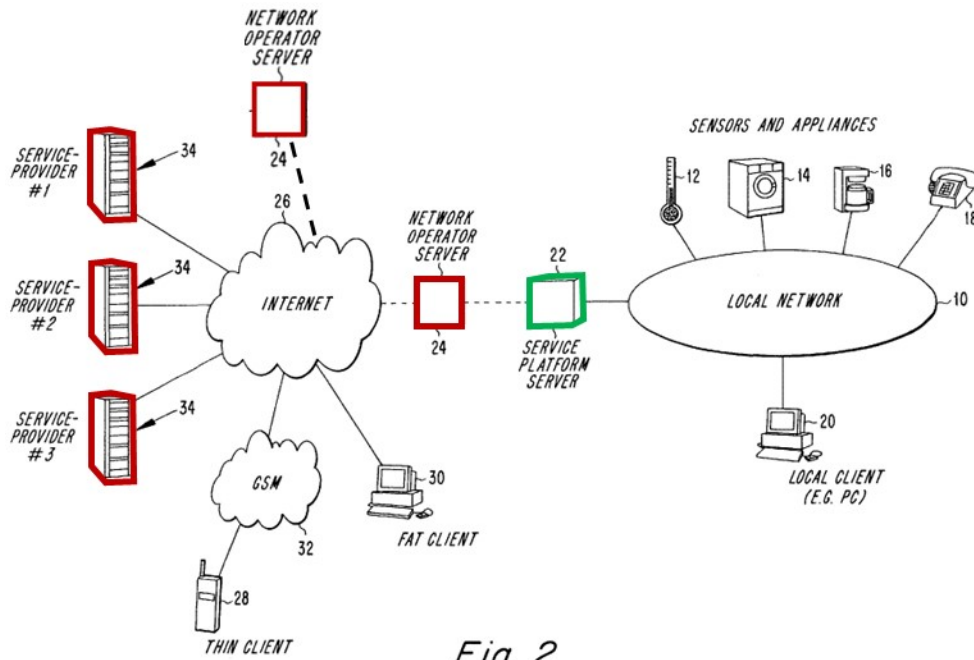


Fig. 2

EX1004, FIG. 2.

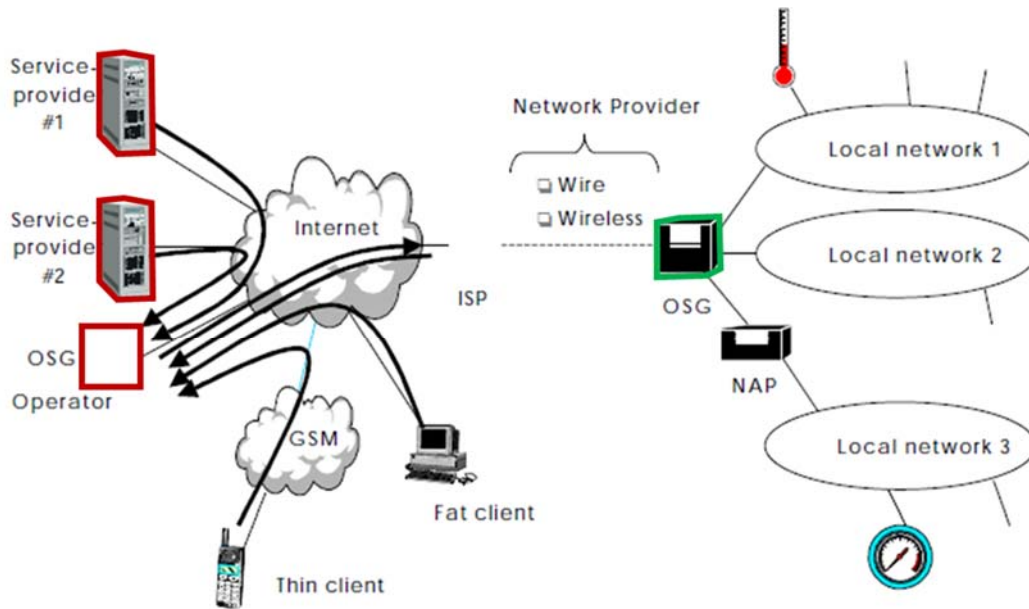
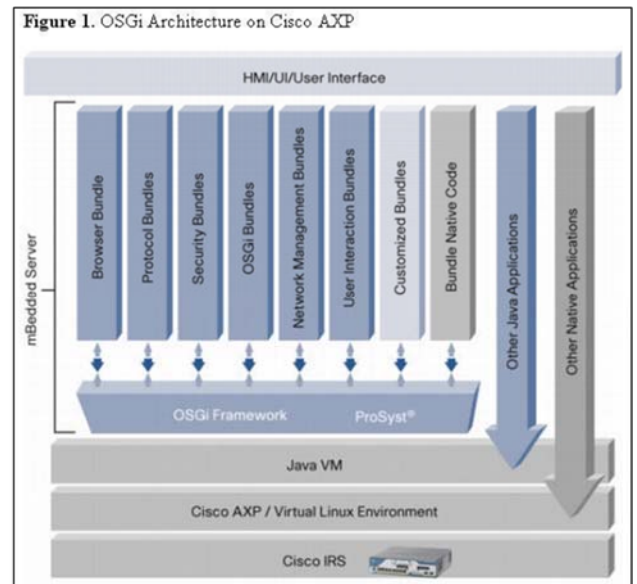
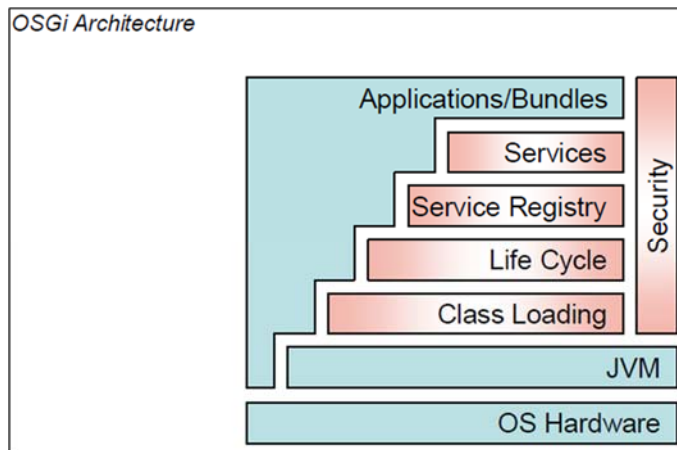


Figure 2. OSGi End to End Architecture

EX1050, FIG. 2; EX1003, ¶192.

The standardized “OSGi Service Platform” was based on “the Java Runtime Environment (JRE), which include[d] the [JVM] that isolates the software component developer from the details of the underlying OS and hardware.”

EX1008, 2, 6-7, 16. The platform was used to “dynamically install and uninstall Java modules, which OSGi call[ed] ‘bundles.’” *Id.* Accordingly, the standardized OSGi architecture was virtualized on device hardware (e.g., router, server), executing in at least one JVM running on the device’s host OS:



EX1013, 3-4; EX1029, 1-2; EX1010, Figure 2; EX1016, 7; EX1003, ¶¶193-194.

The “OSGi **remote management** architecture” included a “**management API**” (the claimed “**application management portal**”) allowing bundle (and updates) installation/uninstallation in remote JVMs, and used “standardized **hot update** technology” allowing runtime updates to multi-bundle applications “without restarting the JVM.” EX1009, 283; EX1013, 7, 10; *id.*, 3-4, 11, 17-18; EX1003, ¶¶199-201.

The OSGi-standardized platform utilized a “**bundle repository**” (the claimed “**application repository**”) containing OSGi bundles developed by the OSGi Alliance and third party developers, and, for OSGi-administrators, provided a “**web based interface**” through which administrators could search for, download, and install bundles. EX1041, 1; EX1013, 7-8, 14-15; EX1051, 1; EX1042, 1-2; EX1003, ¶¶202-205.

OSGi's standardized Enterprise implementation (EX1028) included communications between a locally-installed "bundle" that's exporting a remote service in one OSGi framework instance and [a remotely-installed] bundle that's importing the service in another [OSGi] framework instance" (the claimed "distributed application" with components installed on each device) using well-known protocols (RMI, SOAP, REST) for providing secure communications between components of a distributed application. EX1008, xiv, 249-250, 252-255, 261-263, Figures 10.9-10.10; EX1046, 1-2, 7-8, 13-15; EX1047, 1-2, 4; EX1048, 1-2; EX1053, 3-5, 68; EX1054, 1; EX1055, 1; EX1056, 1; EX1003, ¶¶207-214.

POSITAs understood "the OSGi platform is an ideal platform for cloud computing." EX1008, 250, 269, xiii-xiv; EX1011, 3-4; EX1003, ¶215.

b. Cloud

"Clouds" was the name adopted in the mid-2000s for managed private/public computer "clusters", and were simply the latest, well-known evolution of distributed systems. EX1063, 699-708; EX1052, xxv-xxvi, 19-20, 45, 271-272; EX1003, ¶¶159-162, 171-178, 216.

POSITAs understood "cloud" meant: "set of computers accessed remotely", or "infrastructure comprising a network of interconnected nodes", with the "key concept" being that "the user does not know where the server is located, or where there is one server or many; everything is 'off in the clouds'." EX1033, 94;

EX1052, xxv, 3, 45; EX1030, [0038]; EX1057, 153; EX1003, ¶217. POSITAs understood “cloud computing” meant: “computing operations carried out on servers that are accessed through the Internet” or “applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards.” EX1033, 94; EX1052, 3; EX1003, ¶218.

Cloud deployment models included private, public and hybrid (two or more different types of clouds). EX1052, 3, 7-8; EX1030, [0033]-[0037]; EX1003, ¶219.

The 40+ year-old “virtualization” technology (including as implemented through VMs (*e.g.*, JVMs)) was ubiquitous in **cloud servers** by June 2013. §V.F.1 (EX1031, 569-570; EX1036, 278-279); EX1052, xxvi, 10-12, 19-20, 51-52, 94-95, 100-103; EX1003, ¶220. Having a plurality of VMs running on a single **cloud server** was well-known to POSITAs long before June 2013. *Id.*; EX1030, [0033]-[0037]; EX1052, 69-70, 92; EX1003, ¶221. Indeed, by June 2013, many examples of implementations of a single **cloud server** running a plurality of VMs were known. EX1012, 19-22, FIG. 1; EX1043, 124-125; EX1039, 1-3; EX1003, ¶¶222-225.

It was well-known to utilize “application repositories”/“application stores” (the claimed “application repository”) in cloud-based distributed systems, including those implementing OSGi technology (example depicted below):

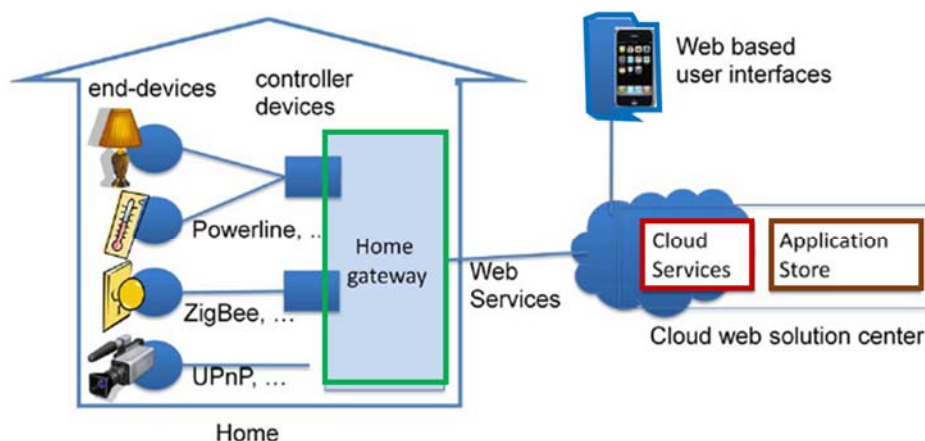


Figure 1: High-level architecture for the smart home

EX1010, 206, 208-209, 212, Figures 1-2; EX1030, [0057], FIG. 4, [0070]-[0074], [0082]-[0084]; EX1042, 1-2; EX1003, ¶¶226-228.

VI. IDENTIFICATION OF CHALLENGES

A. Challenged Claims

This Petition challenges '095 Patent Claims 1-2, 6-8, 10-13, and 15.

B. Statutory Grounds for Challenges

The Challenges are set forth in detail below and summarized as follows:

Ground	Claims	Basis	Reference
1	1-2, 6-8, 10-13, 15	§103 (AIA)	Vasell (EX1004/EX1015/EX1016) in view of Alves (EX1008), Hall (EX1009) and Rellermeyer (EX1011)

Vasell is Patent 6,496,575, issued December 17, 2002 (EX1004), claimed priority to, while expressly incorporating by reference, Provisional Applications

60/088,437 filed June 8, 1998 (EX1015) and 60/123,971 filed March 12, 1999 (EX1016). EX1004, 1, 1:5-10. Accordingly, Petitioner refers to the combined disclosure of EX1004, EX1015 and EX1016 as “Vasell”, and makes each citation to the corresponding Vasell exhibit.

Alves (EX1008) is “OSGi in Depth”, by Alves, published in December 2011, publicly accessible by March 2012, and actually disseminated to interested persons at least by November 2012. EX1014, ¶¶44-65.

Hall (EX1009) is “OSGi in Action”, by Hall, published in April 2011, publicly accessible by December 2011, and actually disseminated to interested persons at least by December 2011. EX1014, ¶¶66-87.

Rellermeyer (EX1011) is “Dependability as a cloud service: a modular approach,” by Rellermeyer, presented at the 2012 IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, published in the conference proceedings by IEEE, in August 2012, publicly accessible by late-September 2012, and actually disseminated to interested persons at least by April 2013. EX1014, ¶¶20-43.

Each reference qualifies as prior art under AIA §102(a)(1), and none of these references were cited nor applied by the examiner during the '095 Patent prosecution.

VII. IDENTIFICATION OF HOW THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Ground 1: Claims 1-2, 6-8, 10-13 and 15 are obvious over Vasell in view of Alves, Hall and Rellermeyer

1. Vasell

Vasell discloses foundational concepts of a **distributed system** architecture, including implementation in a Java run-time environment, that was standardized under OSGi. §V.F.3.a; EX1003, ¶¶179-192, 291. Vasell describes hardware of “the **service gateway system**,” which is its system for processing data packets shown in Annotated FIG. 2 (below). The hardware includes (i) a **service platform server 22** connected to the local area network 10 and connected to (ii) a **network operator server 24** “via the Internet”, which is connected to a (iii) plurality of **service provider equipment 34** “via... Internet 26”:

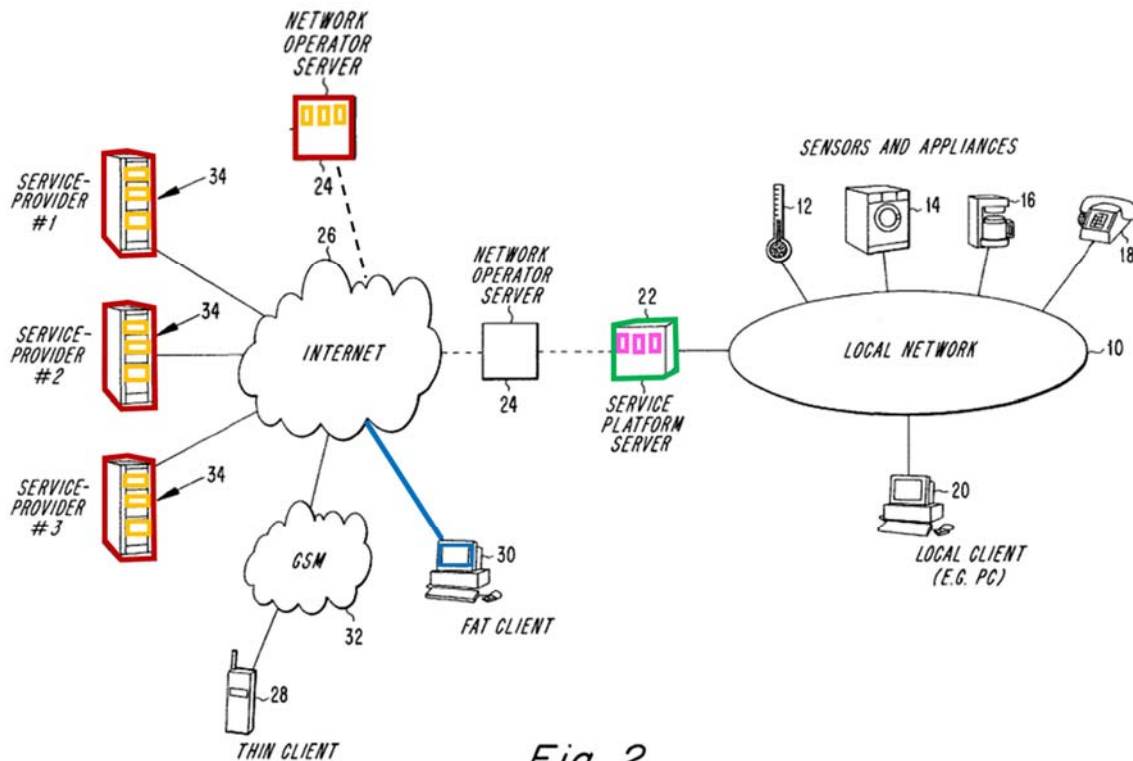


Fig. 2

EX1004, 4:31-47, 5:61-6:20 (“[s]ervice platform server” is “an edge server”), 6:62-7:6 (“[n]etwork operator server...equipment... includes one or more servers”), 7:14-18 (“[s]ervice provider equipment... includes computer systems or servers”), 7:12-14; EX1016, 4, 6-9, FIGS. 2, 5; EX1003, ¶¶294-295, 298-302.

Vasell discloses a plurality of distributed applications—called “service applications”—each having a component installed on a service platform server and a component installed on a network operator server and/or service provider equipment, and each implementing a “service” for end users, which ranged from “household users and residences..., an entire office building, an industrial plant, or even a campus-type organizational facility.” EX1004, 10:32-41; EX1003, ¶306.

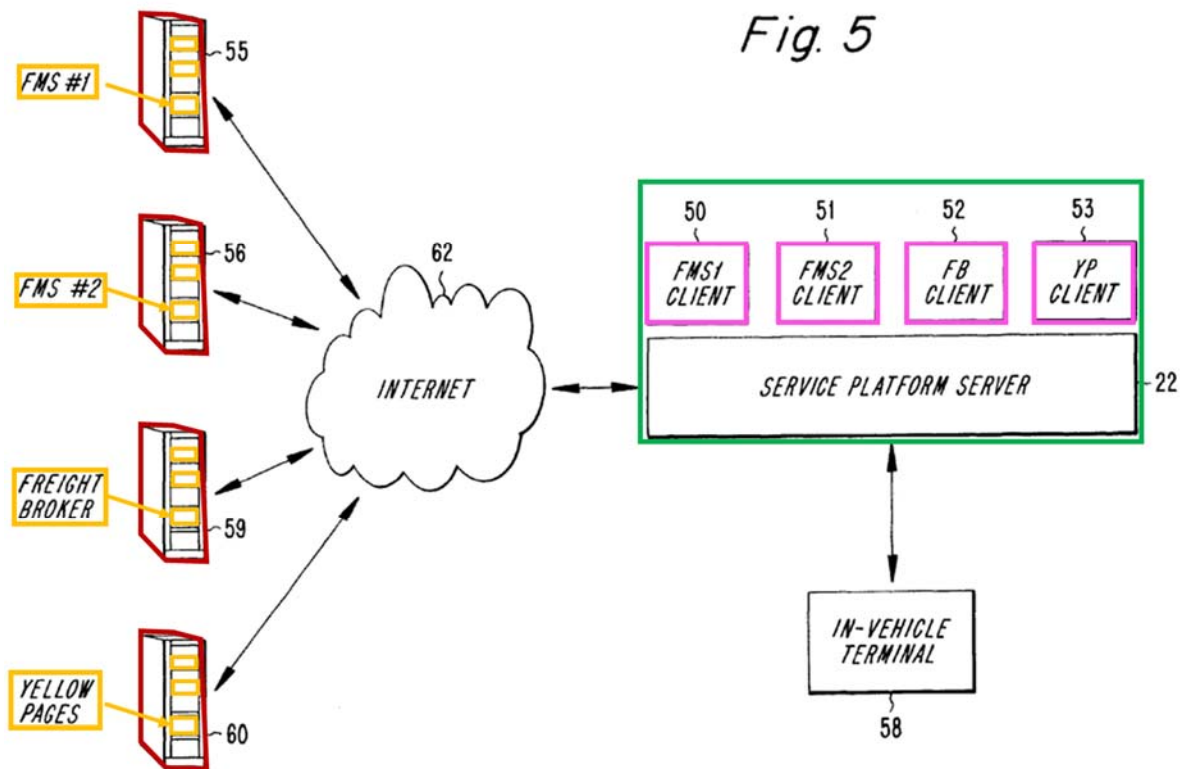
Annotated FIG. 2 shows Vasell's disclosure of a plurality of **service applications** (each including a plurality of **distributed** components), where each **service application** component is respectively hosted on a JVM running on the OS of a **service platform server**, **network operator server**, or **service provider equipment**. *Id.*; EX1004, 2:64-3:1, 8:53-9:18, 24:49-51, 12:27-29, 11:13-58, 18:23-30; EX1016, 4, 7-9; EX1003, ¶¶296-297.

POSITAs understood Vasell's disclosed network of **network operator server(s)** and **service provider servers**, accessed "transparent[ly]" through the Internet, each respectively hosting distributed **service application** components on a JVM running on the OS on the **server's** hardware to implement services through the **service gateway system**, to constitute a "cloud", and each such **server** a programmable **cloud device**. *Id.*; EX1004, 2:60-3:1; §V.F.3.b; EX1003, ¶303.

Vasell discloses "implementation of... services according to the present invention may take advantage of **distributed processing techniques** whereby a service may be implemented using **software** operating on two or more of the servers or processors associated with the [**service gateway system**] architecture illustrated in FIG. 2." EX1004, 8:53-58, 2:60-3:1; EX1016, 4, 7-9. "For example, **a first portion of software** associated with [an] environmental service... may reside... **on the #1 service provider equipment**... **A second portion of the software** used to implement [this service]... may reside **as part of the network**

operator server... Likewise, a **third portion of the software** used to implement this service may reside **on the service platform server**.” *Id.*; EX1004, 8:53-9:18, Annotated FIG. 2, 24:49-51, 12:27-29, 11:42-58; EX1016, 4, 7-9; EX1003, ¶¶304-307.

Annotated FIG. 5 illustrates another example of distributed **service applications**, where each distributed **application** has a **client component** and a **service provider component**. Vasell discloses **mobile service platform server 22** has various “service **software client packages**” (50-53) loaded thereon that communicate with **associated software components** loaded on **service provider servers** (55, 56, 59, 60) to provide services to trucking industry users:



EX1004, 11:11-58, 2:64-66, 18:27-30; EX1016, 4, 7-9; EX1003, ¶¶308-309.

Vasell emphasizes the importance that communications between the respective distributed **service application** components “**be secure**” and expressly envisioned “**widespread** usage of... security techniques **including data encryption... so that the implementation, monitoring... associated with these services is robustly defended against intrusion and manipulation.**” EX1004, 11:64-12:10; EX1016, 3-4, 6-9; EX1003, ¶¶310.

Vasell describes the **service gateway system** as having an “open” **service application** development environment that included “**different** service providers” and “**third party** developers”, and “potentially opens **third-party markets** for... services.” EX1004, 3:10-15, 5:27-33, 10:18-23, 13:37-43, 22:35-37; EX1016, 4, 7-9; EX1003, ¶¶311, 324-325. Accordingly, Vasell discloses that “mechanisms for **authentication, authorization** and access control are integrated within the **service gateway system** design.” *Id.*; EX1004, 4:52-65, 15:46-47.

Vasell discloses its modular, virtualized **distributed system** “ensure[d] the security, isolation and robustness of the **service applications**”, “achieve[d] application integrity”, allowed “multiple **service applications**... to simultaneously operate without detrimentally affecting each other”, and “prevent[ed] operation of a **service application** from disrupting other **service applications**, thus ensuring integrity of the **service gateway system.**” *Id.*; EX1004, 20:49-52, 18:12-13, 4:58-

62, 5:11-15, 18:2-5, 12:47-54, 2:50-53, 3:19-24; EX1016, 4, 6-8; EX1003, ¶¶315-316.

Vasell discloses the **service gateway system** is “**remotely managed**” via “**management system service**” of “system services layer 110”, “implemented in the form of...**boxlets**... executing within cells”, and “provid[ing] **an external interface** through which the **service applications**... [are] downloaded, installed, removed, executed, and controlled”:

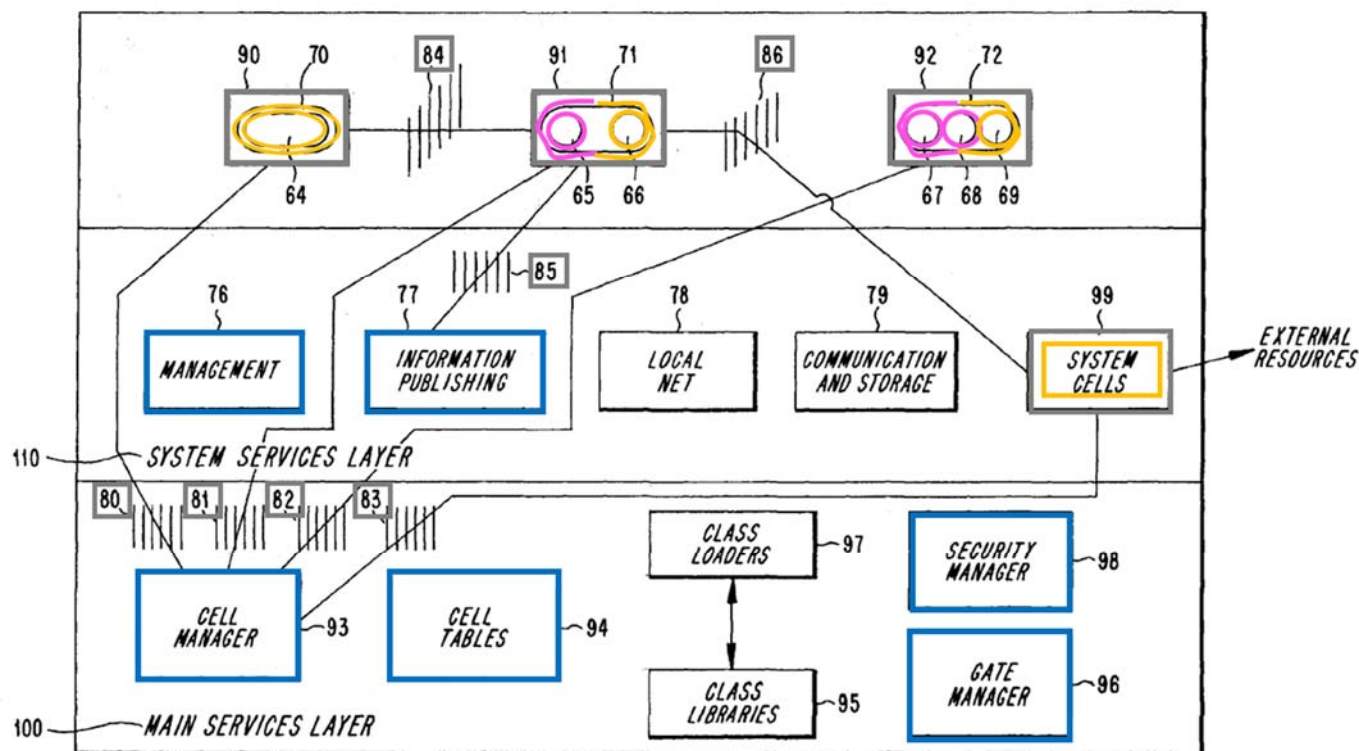


FIG. 6

EX1004, Annotated FIG. 6 (above), 12:11-29, 20:23-31, 22:34-37, 20:10-22:30, 15:58-17:39, 4:44-50, 5:16-26, 9:20-40; EX1016, 4, 6-9, FIG. 4; EX1003, ¶¶317-318. The **management software** includes “**diagnostic**” **software**, running on

network operator server(s) (and/or service provider equipment), to “monitor”, “maintain and control” distributed service applications running on the service gateway system servers. EX1004, 9:20-41, 6:8-20, 3:2-5, 11:40-46; EX1016, 6, 8-9; EX1003, ¶319.

2. Alves

Alves (EX1008) provides an “*in-depth*” description of various technical implementation details of the OSGi specifications (which are implemented in a Java run-time environment). EX1008, xiii-xiv; EX1003, ¶328.

Alves states “the OSGi platform is an ideal platform for cloud computing.” EX1008, 250, 269, xiii-xiv, 261-263, 266-268. Alves identifies that OSGi’s standardized Enterprise implementation (EX1028) included communications between a locally-installed “bundle that’s exporting a remote service in one OSGi framework instance and a [remotely-installed] bundle that’s importing the service in another [OSGi] framework instance” (the claimed “distributed application” with components installed on each device) using well-known protocols for providing secure communications between distributed application components such as RMI, SOAP and REST. EX1008, xiv, 249-250, 252-255, 261-263, Figures 10.9-10.10; *see also* §V.F.3.a (identifying well-known secure implementations of these well-known protocols); EX1003, ¶¶329-330.

Alves discloses that, in a private/community/hybrid cloud OSGi Enterprise implementation, where the network administrator (rather than a cloud provider) continues to “manage OSGi framework instances and their bundles” (e.g., EX1008, FIGS. 10.10, 10.12), a “cloud provider could give [the administrator] access to an OSGi bundle repository (OBR)” in the provider’s cloud, from which the administrator “could install, update, and uninstall bundles as needed”, and for which developers “can design and validate [their] applications using any OSGi implementation in a standalone environment before moving to the cloud.” EX1008, 263, 266; EX1003, ¶¶331-332, 335-336.

Alves discloses the importance of software developers “validat[ing]” (testing and certifying as successfully tested) their applications, e.g., before “moving [them] to the cloud” OBR, and technical implementation details for “effectively” and “eas[ily]” doing such application testing. EX1008, 126-130, 266; EX1003, ¶¶333-334.

Alves also discloses that, in a public cloud OSGi Enterprise implementation, where a cloud provider (and not the network administrator) “manages OSGi framework instances and their bundles” that are “hosted in machines managed by the cloud provider” (e.g., EX1008, FIGS. 10.11, 10.13), well-known “elasticity in the cloud” (the Patent’s “cloud breathing”) is supported by the OSGi platform. EX1008, 266-268. EX1003, ¶337.

3. Hall

Hall (EX1009) “thoroughly describe[s] the important aspects of OSGi and show[s] how to use them” in the standardized Java run-time environment.

EX1009, xx; EX1003, ¶339.

Hall identifies “the OSGi framework is designed to take advantage” of “the built-in support for secure sandboxes” in “the Java platform”, including assigning permissions to “bundles”—called “boxlets within cells” or distributed service application “portions” in Vasell—, and having the AccessController and SecurityManager components of the JVM allow or deny access to bundles based on such permissions. EX1009, 438, 440-448, 464; *see also* §V.F.1; EX1003, ¶340.

Hall explains that digital signatures were another well-known Java security feature “used for OSGi”, especially in environments where bundles (and associated upgrades) are developed by third parties. EX1009, 457-463, 331-334, 471-476; *see also* §V.F.2; EX1003, ¶¶341-342. “In OSGi, the signer of a bundle is associated with it” and, “[w]ith this association,” the OSGi environment management agent “grant[s] permissions to a bundle based on its signers,” *i.e.*, based on the bundle developer’s identity. *Id.*; EX1009, 458-459, 476. Hall explains that this standardized “digital signing”, and authenticity and integrity verification, was applicable to both original, and updated, bundle versions to

render them “tamperproof.” *Id.*; EX1009, 331-333, 179; *see also* §V.F.2; EX1003, ¶¶343-345.

Hall explains that the “JVM security manager”, of the JVM(s) (each of which, in the OSGi environment, is respectively running on a network device’s host OS in which the **bundles** are respectively deployed), performs digital signature matching to determine these **bundle**-specific “permissions”, “control [each **bundle**’s] access to sensitive framework operations and information”, “control[] which services a **bundle** can provide or use”, make **bundle**-specific “deny-access decisions”, determine the local resources “required by the **bundle**”, “enforce” local resource restrictions “at execution time”, etc. EX1009, 471, 439-448; EX1003, ¶¶346-348.

4. Rellermeyer

Rellermeyer discloses a “service... call[ed] **CLOUDDP**... designed to turn OSGi services”—“objects that are specifically shared between [**application**] **components**” called “**bundles**”—into **elastic** units of deployment which can be migrated and replicated across multiple **cloud** nodes to increase the availability and reliability of the administered service.” EX1011, 2-4; EX1003, ¶351. The **CLOUDDP** service “can be architected using features **already provided by** OSGi” because, for example, (i) “the compositional approach of modular software... is key to building scalable and dependable systems across a varying set

of **machines in the cloud**", (ii) OSGi's use of "services... as a design element... result[s]... [in] less entangled and more flexible" systems, and (iii) the "OSGi Enterprise Specifications" already detail "the **API** of the remote service **admin**" ("the **interfaces** through which a possible **topology manager** can interact") and "the mechanism for importing **remote** services into... a **local** OSGi framework and exporting **local** OSGi services" to the **remote** service by creating "a **local** proxy for the **remote** service" and "binding [it] to the **remote** service", which is "reachable through... a known URI or other identifier." EX1011, 2-4; EX1003, ¶352.

"**CLOUDDEP**... takes over the role of a... local topology manager on each **node** of the cloud deployment" and "**monitors** the [**remote**] service invocations" on **local** ("**client**") **nodes**. Rellermeier explains "load-balancing" (Figure 2 (below), "case 1") and "elastic scalability" (below, "case 3") usage cases, which provide the '095 Patent's "cloud breathing" functionality:

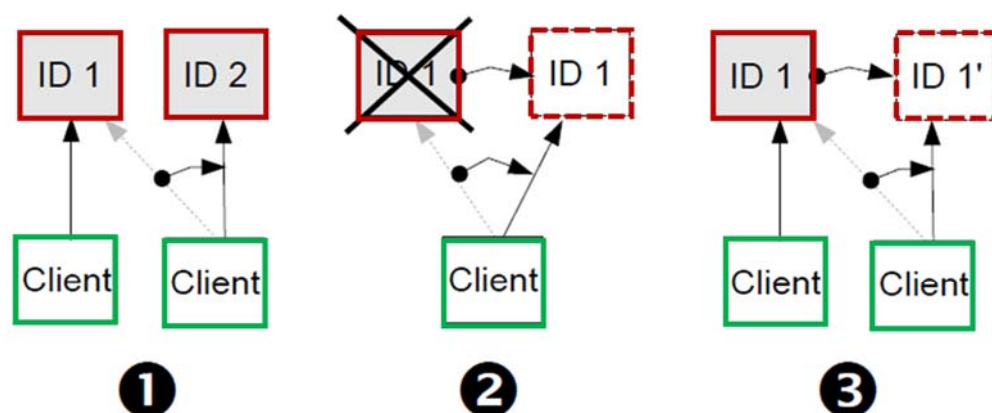


Figure 2. Different Use Cases for a Dependability Service. 1: Load balancing, 2: Failover, 3: Elastic Scalability.

EX1011, 4-5; EX1003, ¶353. In “load balancing” Case 1, “**CLOUDDP**... instrument[s] the **local** proxy to gather information about the frequencies of [remote] service invocation and response times”, and “[i]n the event that one **remote** service **instance** [“ID1”] **becomes overloaded**, detectable by an increase of invocations and an increase of the response time, **CLOUDDP** can re-bind a **client** to a different [remote] service **instance** [“ID2”], thereby re-distributing the load among the [remote] service **instances**.” EX1011, 4-5; EX1003, ¶354. In “elastic scalability” Case 3, “[i]n the event of a [remote] service **over-utilization**, **CLOUDDP** can start a new **instance** of the [remote] service [“ID1”] and bind some of the **clients** to the new service” and “[s]cal[es] back to a smaller number of [remote] service **instances**... by... moving all **clients** of a [remote] service to a different existing **instance** [“ID1”] and then dropping the service **instance** [“ID1”].” EX1011, 5; EX1003, ¶355.

5. The Motivation to Combine

As detailed in State of the Art §V.F.3.a, Vasell’s architecture and methods are foundational concepts (including Java run-time environment implementations) that were eventually standardized under OSGi. Technical details intended to implement Vasell’s teachings are described in each of the Ground 1 published references—Alves, Hall and Rellermeier. Accordingly, the motivation for POSITAs to modify Vasell’s teachings, to include the features described in these

references for their intended purpose and to achieve predictable benefits with a reasonable expectation of success is demonstrated by the references themselves. EX1003, ¶¶377-378.

For example, Vasell discloses a **service gateway system** where connectivity-based services are implemented using **service applications** whose processing is **distributed** across respective Java components, respectively executing in JVMs hosted on **local** and **remote (cloud)** servers' OSs and securely communicating with each other, and in which many different service providers and third parties respectively develop the distributed **service applications**, and their underlying Java components, for many different connectivity-based services. Network administrators remotely, via a **management interface**, and **management software modules** also executing on the **service gateway system servers**, **manage** the downloading, installing, upgrading and termination of these distributed **service application** components, and monitor and control their implementation on the **service gateway system servers** (including dynamic resource allocation). §VII.A.1; EX1003, ¶379.

POSITAs would be motivated to modify Vasell's teachings with a **cloud-provider-application-repository** storing distributed **applications**, for which developers can design and validate their **applications** using any OSGi implementation in a standalone environment before moving to the cloud, as Alves

teaches, because it provides increased flexibility and security benefits. EX1008, 263, 266. Specifically, Alves describes technical details for (i) implementing OSGi-compatible **distributed systems** in various well-known cloud environments (including utilizing a **cloud-provider-application-repository** to store “validate[d]” (tested and certified as successfully tested) **bundles**), and (ii) using well-known protocols to provide secure communications between **locally-** and **remotely-** installed components of distributed **applications** implementing services. §VII.A.2; EX1003, ¶¶380-381.

POSITAs would be motivated to modify Vasell’s teachings with **management software** that requires digital signatures for distributed **applications**, as Hall teaches, to provide increased security for using third-party-developed applications. Specifically, Hall discloses technical details for having the OSGi **management software**, and each “JVM security manager” of the JVMs (each respectively running on a network device’s host OS in which the **bundles** are respectively deployed), implement well-known, standardized digital signature techniques to verify the respective authenticity and integrity of each downloaded **bundle** (including updated versions). §VII.A.3; EX1003, ¶¶382-383.

POSITAs would be motivated to modify Vasell’s teachings with **management software** that provides for load balancing, failover and elastic scalability as Rellermeyer teaches. Specifically, Rellermeyer discloses a

management service that, “using features already provided by OSGi” specifications, has the OSGi management software monitoring and controlling automatic load (resource usage) expansion/contraction across multiple cloud servers. §VII.A.4; EX1003, ¶384.

POSITAs understood that the beneficial aspects that Alves, Hall and Rellermeyer described for OSGi distributed systems—increased scalability, having a cloud-provider-application-repository storing distributed software application components (and associated updates) developed by different application developers, increased security, capabilities of granting application-specific-permissions based on the respective application developer’s identity, and capabilities of elastically deploying cloud resources to increase the availability and reliability of services—would be identically beneficial to Vasell’s foundational service gateway system. EX1003, ¶385; *see also* State of the Art §V.F.1-§V.F.3.b.

POSITAs would be motivated to modify Vasell’s teachings to utilize a cloud-provider-application-repository of “validate[d]” applications in a cloud environment, implement inter-application communication protocols with well-known secure communication capabilities, implement well-known, standardized digital signature techniques, and to have its remote manager monitor and control load expansion/contraction across multiple cloud servers, in order to achieve the known and predictable benefits associated therewith. *Id.*; EX1003, ¶386.

In view of this section (and discussed further below in the claims analysis),
Petitioner has demonstrated at least:

- i. Vasell and Alves themselves provided the motivation that would lead POSITAs to implement Vasell’s network of **service provider servers** and **network operator server(s)**, as a “cloud” using a **cloud-provider-application repository**, to implement “many different types of services” using “validate[d]” components of distributed **applications** developed in an “open” environment including “different service providers” and “third party developers.” EX1003, ¶388.
- ii. The general knowledge in the art was that Enterprise OSGi’s and Vasell’s shared features—*e.g.*, modular software **applications**; **distributed** execution of **service applications** in a two-tier, multiple machine (**local** and **remote (cloud) servers**), virtualized (JVM) architecture; dynamic **service application** component updates without affecting the concurrent execution of **service applications** on the same **servers**; and dynamic resource usage (load) monitoring and control—made OSGi (and Vasell) an “ideal” system to implement in a well-known “cloud” environment (including a private/community/hybrid cloud implementation utilizing a **cloud-provider-application-repository**). EX1003, ¶¶389-393.

- iii. The beneficial aspects of having a **centralized repository** for **software** developed by different application developers, and requiring each developer to “design and validate” its **software** in its own “standalone environment before moving” “validate[d]” **software** to the **cloud-provider-application-repository**, as Alves teaches, would be accomplished as a mere application of a known centralized **repository** solution to a known **distributed system** ready for improvement to achieve known and predictable results. EX1003 ¶¶394.
- iv. Vasell and Hall themselves provided the motivation that would lead POSITAs to implement Vasell’s administrator, and **management software**, **managed** upgrades of Java components of distributed **service applications**, in Vasell’s “open” development environment, using, as Hall teaches, standardized digital signature techniques in OSGi’s JVM environment (shared with Vasell) to respectively verify the authenticity and integrity of each such component, and achieve well-known security benefits with a high likelihood of success. EX1003, ¶¶395-396.
- v. Vasell and Rellermeyer themselves provided the motivation that would lead POSITAs to improve, “using features already provided by OSGi” specifications and with the knowledge that OSGi’s and

Vasell's shared features made OSGi (and Vasell) an "ideal" system to implement in a well-known "cloud" environment, Vasell's remote manager monitoring and dynamically allocating service gateway system resources using Rellermeyer's teachings of the OSGi management software monitoring and controlling load expansion/contraction across multiple cloud servers, with confidence that the combination would have a high likelihood of success. EX-1003, ¶¶397-398.

6. Detailed Application of Vasell, Alves, Hall and Rellermeyer to the Challenged Claims

Claim 1

[1.0] A system for processing data packets in a network, the system comprising:

To the extent the preamble is limiting, Vasell discloses this limitation because it discloses a "service gateway system" that utilizes "distributed processing" across "servers or processors" connected to each other over "an Internet Protocol (IP) network such as the Internet" (*system for processing data packets in a network*). The service gateway system is shown in Vasell's Annotated Figure 2 (below), which includes hardware: (1) a "service platform server 22" "connected to the local area network 10" and connected to (2) a "network operator

server 24 via the Internet”, which is connected to (3) a plurality of service provider equipment 34 “via... Internet 26”:

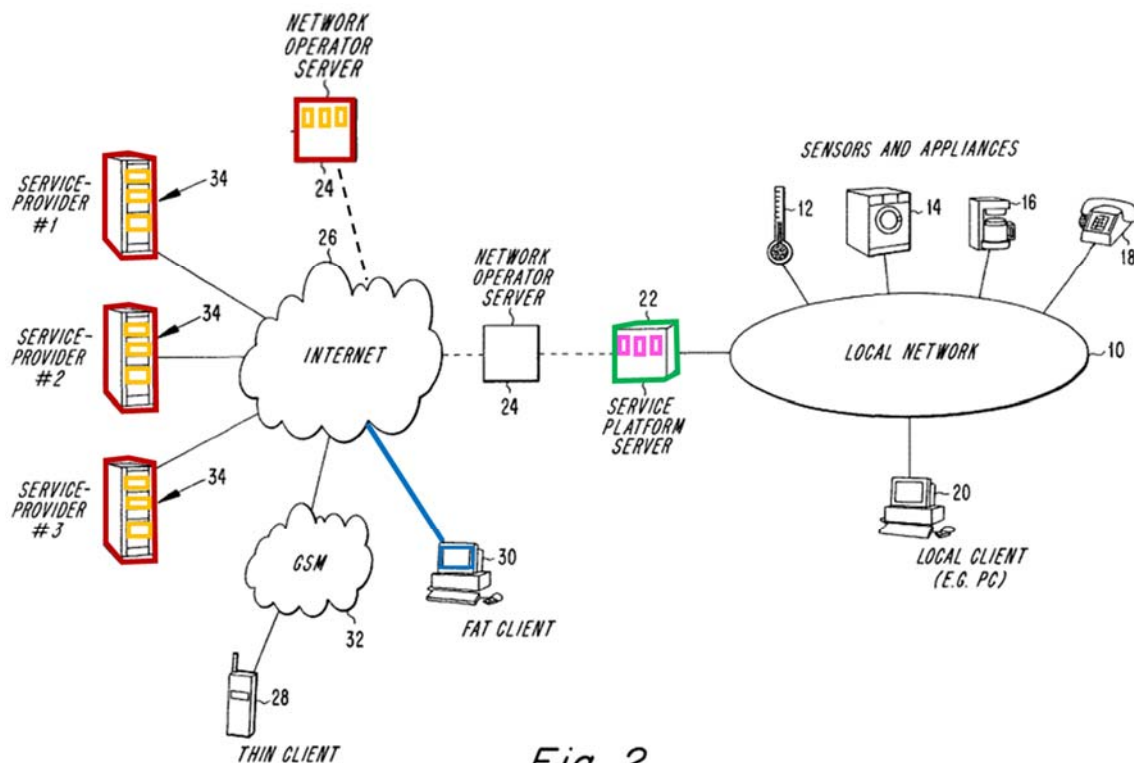


Fig. 2

EX1004, 4:31-47, 5:61-6:20, 6:62-7:6, 7:12-18; EX1016, 4, 6-9, FIGS. 2, 5;
EX1003, ¶¶400-401.

Vasell discloses that the distributed “system software”, of each of the service gateway system servers, includes “a Java runtime in the form of a Java virtual machine (JVM)” and an OS hosting the JVM, which could be “[a]ny UNIX-like standard” OS, provided it has “fairly strong-resource control and protection mechanisms to ensure application integrity.” EX1016, 4, 6-9, FIGs. 2, 5; EX1004, 18:27-30; EX1003, ¶¶404-405.

Vasell discloses the **service gateway system** “provides a platform for connectivity based services”, where each of a plurality of distributed **software applications**—called “**service applications**”—implement a respective “service” for end users. Each “**service application** may be **distributed** among various pieces of equipment which are geographically separated”, such that each distributed **service application** has a **component** installed on a **service platform server** and a **component** installed on a **network operator server** and/or **service provider equipment**. EX1004, 4:31-50, 2:64-3:6, 1:41-45, 8:53-9:18, 12:21-33, 11:11-58; EX1004, 4, 7-9; EX1003, ¶¶402-403.

Accordingly, the above-annotations in Annotated FIG. 2 show a plurality of **service applications** (each including a plurality of **distributed** components). Specifically, each component of a **service application** is respectively hosted on a JVM running on the respective OS of the **service platform server**, the **network operator server(s)**, and the **service provider equipment**. *Id.*; EX1004, 4:31-47, 5:56-6:20, 6:62-7:14; EX1016, 7-9; EX1003, ¶406.

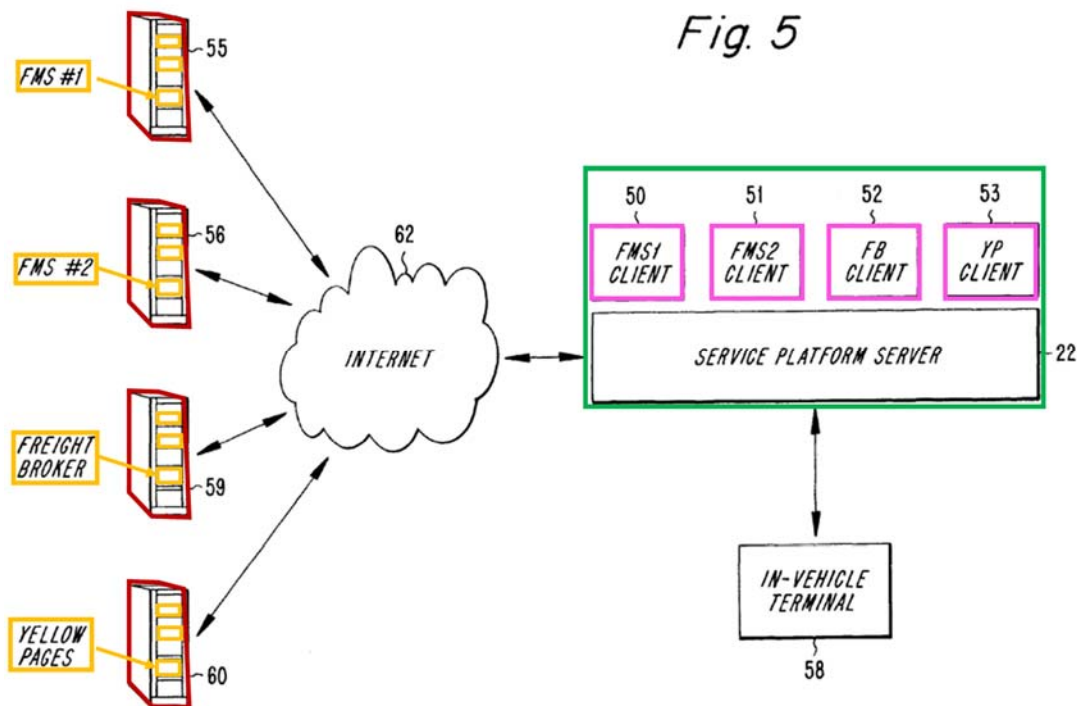
[1.1] a programmable network device hosting a plurality of first network applications;

Vasell discloses this limitation. Vasell discloses “**service platform server**” [*programmable network device*] is “an edge server” and “the hardware portion of a service gateway or services platform”, and states **it** “may provide service support for” “household users and residences..., an entire office building, an industrial

plant, or even a campus-type organizational facility.” EX1004, 5:61-6:20, 6:45-49, 10:32-41; EX1003, ¶¶408-409.

Vasell discloses “service platform server... may contain... part of... software... applications... used to implement one or more connectivity based services which have been requested and/or subscribed to by the end user” [*hosting a plurality of first network applications*]. EX1004, 6:21-25, Annotated FIG. 2; EX1003, ¶410.

Vasell discloses several examples of “distributed” service applications in the service gateway system, and of “portions”/components thereof hosted on service platform server. *Id.*; EX1004, 9:42-61, 8:53-9:18. For example, Annotated FIG. 5 illustrates “client software packages 50-53 which are running at any given time on the service platform server”:



EX1004, 11:11-39, 13:28-36; EX1003, ¶¶411-415.

[1.2] a programmable cloud device hosting a plurality of second network applications,

Vasell alone, in view of Alves or further in view of Rellermeier, discloses this limitation.

Vasell discloses “**service provider equipment**... include[s] computer systems or servers, may be the initial repository for **software** used to implement connectivity based services” [*programmable cloud device*]. EX1004, 7:4-20, 11:40-46, 12:31-33; EX1003, ¶417.

Vasell discloses “**network operator server**... include[s] one or more servers, i.e., relatively powerful computers operating in conjunction with one or more

secondary storage devices” [another *programmable cloud device*], and “handles control for the operation, maintenance and support of the service[s] implementation.” EX1004, 6:45-7:3, 6:16-20, 9:25-40, 11:40-46; EX1016, 5-9; EX1003, ¶¶418-420.

Moreover, Vasell discloses the **service provider equipment** of each service provider, and **network operator server** of each network operator, may each host respective “**portions**”/**components** of “distributed” **service applications** to implement each of a respective plurality of services [*programmable cloud device hosting a plurality of second network applications*]. EX1004, 8:53-9:24, Annotated FIG. 2, 2:64-66, 11:12-46, Annotated FIG. 5, 13:28-36, 11:47-58 (example of same **service provider** providing plural services), 12:27-29; EX1016, 4-5, 7-9; EX1003, ¶¶422-425.

Accordingly, POSITAs understood, as explained in the State of the Art §V.F.3.b, Vasell’s disclosed network of **network operator server(s)** and **service provider servers**, accessed “transparent[ly]” through the Internet, each respectively hosting “distributed” **service application** “portions”/components on a JVM running on the OS on the **server’s** hardware to implement services through the distributed **service gateway system**, to constitute a “*cloud*”, and each such **server** a *cloud device*. EX1004, 2:60-66; §V.F.3.b (EX1033, 94; EX1052, xxv, 3, 45-46, 271-272; EX1030, [0038]; EX1057, 153); EX1003, ¶421.

To the extent PO asserts that Vasell does not explicitly disclose its **service provider servers**, and **network operator server(s)**, as “*cloud*” devices, as discussed in the Motivation to Combine §VII.A.5, it would have been obvious to POSITAs to implement these servers as a “cloud” as a mere design choice with a reasonable expectation of achieving well-known benefits. EX1003, ¶426; §VII.A.5.

As discussed in the Motivation to Combine §VII.A.5, POSITAs would be motivated to select the known design choice of implementing Vasell’s **service provider servers**, and **network operator server(s)**, in, for example, a private/community/hybrid cloud as Alves teaches, where the **service gateway system** remains **managed** by the network operator as Vasell discloses, with a reasonable expectation of achieving known and predictable benefits, such as rendering “[t]he **system**... more scalable, because... we have a flexible setup, where we understand which **components** can be replicated and how best to partition the requests”, because POSITAs understood that Vasell’s disclosed features (shared with OSGi) made Vasell’s **service gateway system** an “ideal” system to implement in a cloud environment. §VII.A.5 (EX1008, xiii-xiv, 250, 261-263, 266-268, FIGS. 10.10-10.13); EX1003, ¶¶428-430.

Furthermore, to the extent PO asserts that, (i) in order to be a programmable “cloud” device, Vasell’s **remote management** functionality must include monitoring and control functionality for load (resource usage) expansion and

contraction across Vasell's **service provider servers** and **network operator server(s)**, and (ii) Vasell in view of Alves does not disclose *how* this is explicitly accomplished, for the reasons discussed in the Motivation to Combine §VII.A.5, POSITAs would have been motivated to modify Vasell's teachings with Rellermeyer's teachings of providing load balancing and elastic scalability "using features already provided by OSGi" specifications. §VII.A.5 (EX1008, 266-268; EX1011, 2-5); EX1016, 9; EX1003, ¶¶431-433.

[1.3] wherein at least one of the plurality of first network applications in the programmable network device and at least one of the plurality of second network applications in the programmable cloud device are in secure communication with each other to form a distributed application;

Vasell alone, or in view of Alves, discloses this limitation. Vasell discloses **distributed service applications**, implementing connectivity-based services in the **service gateway system**. EX1004, 8:53-58, 2:60-3:1; EX1003, ¶¶434-435.

As an example of **distributed service applications** in the **service gateway system**, Vasell discloses, referencing Annotated FIG. 2, that "a **first portion** of **software** associated with an environmental service may reside on the **#1 service provider equipment 34**. A **second portion** of the **software** used to implement [this service]... may reside as part of the **network operator server 24**. Likewise, a **third portion** of the **software** used to implement this service may reside on the **service platform server 22**." [*at least one of the plurality of first network applications in*

the programmable network device and at least one of the plurality of second network applications in the programmable cloud device... form a distributed application]. EX1004, 8:53-9:24, 2:64-3:1, 24:49-51, 11:47-58 (example of same **service provider** providing plural services), 12:27-29; EX1016, 4, 7-9; EX1003, ¶¶436-438.

In other words, Vasell discloses a plurality of “distributed” **software applications**—called “**service applications**”—each implementing a “service” for end users, and each having (i) a **component** (“**portion**”) installed on a **service platform server**, and (ii) a **component** (“**portion**”) installed on a **network operator server** and/or **service provider equipment**. *Id.*

Vasell’s Annotated FIG. 5 illustrates another example of “distributed” **service applications** in the **service gateway system**, where each distributed **application** has a **client-hosted component** and a **service-provider-hosted component**. Vasell discloses, using a trucking-industry-services example, **mobile service platform server 22** (“outfitted” on a truck) has various “service **software client packages**” (50-53) loaded thereon communicating with **associated components** loaded on **service provider servers** (55, 56, 59, 60). EX1004, 11:13-46, EX1016, 4, 9; EX1003, ¶¶439-442.

POSITAs understood that Vasell also discloses that the underlying **components** that collectively form each of these examples of “distributed” **service**

applications are in secure communication with each other for several reasons.

EX1003, ¶443.

First, per [1.0], Vasell discloses that each distributed *service application* component (“portion”) is respectively hosted on a “Java Virtual Machine (JVM)” running on the respective OS of the *service platform server*, *network operator server(s)*, and *service provider equipment*, in the *service gateway system*. EX1016, 4, 7-9; EX1004, 18:27-30; EX1003, ¶¶444-445.

Second, Vasell emphasizes: “[S]ecure implementation of... connectivity based services [is] important” and, “[s]ince various service providers, network providers and, possibly, end users, will share infrastructure, it is important that information and **communication associated with each implemented connectivity based service**”—*i.e.*, communications between the respective components collectively forming each of these “distributed” *service applications* to implement each such service—“**be secure.**” EX1004, 11:59-64; EX1003, ¶¶446-447. Specifically, Vasell expressly envisions “**widespread usage of... security techniques including data encryption... so that** the implementation, monitoring... associated with these services **is robustly defended against intrusion and manipulation.**” *Id.*; EX1016, 3, 6.

Third, consistent with the well-known sandboxing of the run-time environment of JVMs running on the host OSs of *service gateway system servers*

(*see* §V.F.1), Vasell discloses that each underlying Java component (“boxlet[s]” within “cell[s]”) collectively forming each distributed **service application** is “designated into various categories having different degrees of access restriction”, where “access rules” (**administrator-defined** for each application’s cell and for application cell “groups”) are **controlled** and **enforced** using, in part, “inter-cell communication... controlled interfaces” called “gates” (**managed** by “**gate manager**”) for **application component-to-component** communications. *Id.*; EX1004, 11:64-12:10, 4:58-65, 13:28-36; EX1016, 8, Figure 4; EX1003, ¶448.

Accordingly, although Vasell uses different words than the ’095 Patent, POSITAs understood Vasell discloses its distributed **service applications** being formed by the respective underlying components securely and “transparent[ly]” communicating with each other using well-known “data encryption” protocols, managed “gates” enforcing cell-specific “access rules”, and Java APIs from the respective virtualized, sandboxed, and “higher level of abstraction” environment of respective JVMs running on the respective OSs of the **service platform server** or **remote (cloud) server** on which they are hosted [*in secure communication with each other to form a distributed application*] with the same, or an equivalent, structure as the ’095 Patent. *Id.*, *see* §V.C; EX1003, ¶¶449-451.

To the extent PO asserts that Vasell does not explicitly disclose **secure** communications between the underlying components of Vasell’s distributed

service applications, POSITAs would be motivated to securely implement Vasell's inter-application communications as discussed in the Motivation to Combine section. Indeed, as discussed in the Motivation to Combine §VII.A.5, POSITAs would be motivated to use well-known protocols (e.g., RMI, SOAP, REST) as Alves teaches, which POSITAs understood, as explained in the State of the Art §V.F.3.a, would provide secure communications between Vasell's Java components of distributed service applications, respectively running in a virtualized (JVM) environment on a host OS of a service gateway system server, and with a reasonable expectation of achieving the increased security benefits expressly disclosed by Vasell. *Id.*; §VII.A.5 (EX1008, 249, 252-255, 261-262, Figure 10.9); §V.F.3.a (identifying well-known secure implementations of RMI, SOAP, REST); EX1003, ¶¶452-455.

[1.4] the programmable network device and programmable cloud device are each powered by a sandboxing operating system which facilitates deployment of the plurality of first and second network applications and

Vasell discloses this limitation. Per [1.0]-[1.3], Vasell discloses the distributed “service applications” as being “modularized”, with each distributed service application component respectively hosted on a JVM running on an OS on either a service platform server or remote (cloud) server, and where each OS hosting each JVM was a “UNIX-like standard [OS]” provided it has “fairly strong-

resource control and protection mechanisms to ensure application integrity” such that “errors in software developed by different service providers” cannot result in the “applications... interfer[ing] with one another.” EX1016, 4, 6-9, FIGS. 2, 5; EX1004, 3:18-23, 18:27-30.

Thus, POSITAs understood Vasell discloses **service platform server** [programmable network device], **network operator server(s)**, and **service provider equipment** [programmable cloud device], are each powered by a sandboxing operating system (e.g., JVM respectively hosted on server OS with “fairly strong-resource control and protection mechanisms to ensure application integrity”, where application-cell-specific-“access rules” are controlled and enforced using “class loaders”, “class libraries”, and “gates”, managed by a “security manager”, “gate manager”, and “cell manager” using application-cell-specific-access-information stored in a “cell table”) (including for **application-component-specific server** resource control). *Id.*; EX1004, 3:19-24, 12:21-29, 13:64-14:7, 15:41-46, 17:40-45, 18:9-19, 18:23-30, 19:42-43, 19:63-20:2, 22:8-18; EX1016, 7-8; EX1003, ¶¶456-462, 464.

Unsurprisingly, given the well-known nature of sandboxing OSs to POSITAs including of Vasell’s disclosed JVMs hosted on server OSs (*see* State of the Art §V.F.1), Vasell’s structure to carry out the [1.4]-recited functions is the PO-indicated structure in the ’095 Patent (§V.C) or equivalent thereof, which states

the sandboxing OS may be “built based on an existing sandboxed operating system.” *Id.*; EX1001, 10:19-25; §V.F.1; EX1003, ¶¶462, 465-466, 470-473, 132-136.

Moreover, Vasell confirms that the sandboxing OS powering each of its **servers** *facilitates deployment of the plurality of first and second network applications* (which collectively form “distributed” **service applications**) by: “ensur[ing] the security, isolation and robustness of the **service applications**”, “achiev[ing] **application** integrity”, allowing “multiple **service applications**... to simultaneously operate without detrimentally affecting each other”, “prevent[ing] operation of a **service application** from disrupting other **service applications**, thus ensuring integrity of the **service gateway system**”, and enabling the implementation of services by the **service gateway system** to “continue undisturbed while [**service application**] configuration updates take place.” EX1004, 20:49-52, 18:12-13, 4:58-62, 5:11-15, 18:2-5, 12:47-54, 2:50-53, 3:19-24, 22:6-30, 1:6-10; EX1016, 4, 6, 8; EX1003, ¶¶468-469, 474.

[1.5] and [the sandboxing operating system (see [1.4])] facilitates upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device;

Vasell alone, or in view of Alves, discloses this limitation. Vasell discloses that the sandboxing OS powering each of the **servers** in the **service gateway system**

(see [1.4]) also *facilitates upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device.*

For example, Vasell discloses “upgrade software” as a “remote... management” operation in which servers are “switched on the fly” to the upgraded “service application”, and describes an example of how the “cell manager” and “cell table” enable an “update to... a new configuration of one of the service applications”, even “if it is running”, such that the “service” provided by this one service application in the service gateway system “may continue undisturbed while [these] configuration updates take place.” EX1004, 5:16-26, 22:6-30, 12:11-29, 2:56-59; EX-1016, 4, 6; EX1003, ¶478. Moreover, POSITAs understood that Vasell’s disclosed structure to perform this function is the PO-indicated structure in the ’095 Patent (§V.C), or equivalent thereof, regardless of whether the algorithm is one at EX1001, 11:29-32, 19:38-20:3, or 20:5-44. *Id.*; EX1003, ¶¶477-489.

Moreover, POSITAs understood, from the same disclosures (see [1.4]-[1.5]), that Vasell discloses that the sandboxing OS powering each of Vasell’s service gateway system servers *facilitates* the implementation of these “hot upgrade” features to update Vasell’s service applications with substantially no interruption to operation of such servers. *Id.*

To the extent PO asserts Vasell does not explicitly disclose its “upgrade software” teachings as applying to the **remote server** (24, 34) “portions”/components of its **service applications**, for at least the reasons discussed in the Motivation to Combine section, POSITAs would be motivated to modify Vasell’s teachings with Alves’ teachings to have Vasell’s **remote management services** download **service applications** (and updates) from a **cloud-provider-application-repository** for installation in the **service platform server(s) and remote (cloud) servers** of Vasell’s **service gateway system** without having to restart the respective JVMs of such **servers**, and with the services “continu[ing] undisturbed while [**service application**] configuration updates take place.” *Id.*; §VII.A.5; EX1003, ¶490.

[1.6] wherein the programmable network device verifies authenticity of the upgrades to the plurality of first network applications and the programmable cloud device verifies authenticity of the upgrades to the plurality of second network applications and wherein the verification is based on unique security keys associated with each of the plurality of first and second network applications;

Vasell alone, or in view of Hall, discloses this limitation. Vasell discloses that authentication of upgrades to any **service application** is important due to the “open” nature of the **service gateway system**. Specifically, Vasell discloses that the “open” **service gateway system** “allow[s] for the development of **service applications** by **third party developers**” using “standard communication protocols

and programming languages”, permits “a plurality of service providers [to] share the same platform infrastructure”, and “opens **third-party markets** for... services.” EX1004, 3:10-19, 5:30-33, 10:18-23, 2:60-66, 13:37-43; EX1016, 4, 8-9; EX1003, ¶¶491-493.

Moreover, Vasell discloses that, because “the **service gateway system** is accessed by and shared with multiple service providers and service gateway users”, “mechanisms for **authentication**... are integrated within the **service gateway system** design”, and “various **authorization and identity verification techniques**... [are] provided so that the implementation... associated with these services **is robustly defended against intrusion and manipulation.**” *Id.*; EX1004, 4:52-65, 11:59-12:10, 15:46-47; EX1016, 3-4, 6; EX1003, ¶¶494-495.

Unsurprisingly, Vasell does not disclose the details of its **service gateway system servers** [*programmable network device* and *programmable cloud device*] implementing well-known “authentication”, “authorization”, and/or “identity verification” security techniques for its “open” development environment of received **service application software** by “different service providers”, “third party developers”, and “nonspecialists”, especially where Vasell discloses that such distributed **service applications** are “Java **applications**” implemented in “a Java runtime in the form of a... JVM” environment for which, as discussed in the State

of the Art §V.F.2, such security techniques were widely implemented by recipient devices. *Id.*; §V.F.2; EX1003, ¶¶496, 135, 138, 142, 145-147.

Indeed, as described in the State of the Art §V.F.2, standardized security schemes that were widely implemented by application developers and implementers for JVM-hosted applications, in this precise environment of developers with varying and unknown trustworthiness levels, required each developer to, using their unique, trusted-authority-issued private keys, “digitally sign” each of their respectively developed applications (and corresponding updates). *Id.* Then, prior to installation of each application/update, a **management agent** (and each “JVM security manager” of the JVMs in which the bundles are deployed on the **servers’** OSs) use the corresponding unique public key to verify the respective developer is authentic and the respective software/update is unmodified. *Id.*; EX1003, ¶503. This well-known structure to perform this function is also the PO-indicated structure in the ’095 Patent (§V.C) or equivalent thereof. *Id.*; EX1003, ¶¶505-512, 135-149.

Thus, POSITAs understood that Vasell discloses this limitation based on its concerns regarding the authentication of upgrades to its distributed Java **applications**, and its use of JVMs hosted on respective OSs of its **service gateway system servers**. *Id.*; EX1003, ¶¶505-506.

Moreover, to the extent PO asserts that Vasell does not explicitly disclose this limitation, it would have been obvious to modify Vasell’s teachings with Hall’s teachings of requiring OSGi-application-developers to “digitally sign” their respectively developed distributed **application components**, including updated versions, as discussed in the Motivation to Combine section, to achieve the benefits of rendering each **application** “tamperproof”, determining the application-cell-specific-access-levels granted by Vasell’s **management services** based on the respective developer’s identity (signature), and to control the **application**’s access to operations, information, services and **server** resources. §VII.A.5 (EX1009, 457-463, 330-334, 472-476, 179); EX1003, ¶¶513-514.

[1.7] an application repository storing distributed applications which have been tested for installation in the programmable network device and programmable cloud device; and

Vasell in view of Alves discloses this limitation. Vasell discloses **management services** that provide “an **external interface** through which **service applications** ... may be downloaded, installed...” [*application management portal* of [1.8]], and the **service application** development environment of the **service gateway system** as an “open” environment, including “**different** service providers”, “**third party** developers” and “nonspecialists”, and “**third-party**

markets for... services.” EX1004, 3:10-15, 5:27-33, 10:18-23, 13:37-43, 22:35-37; EX1016, 4, 7-9; EX1003, ¶¶515-516.

Although Vasell identifies “**service provider equipment**... **may** be the initial repository for **software** used to implement connectivity based services” of such service provider, and “**network operator server**... may need to **receive some portions** of the **service application software** from the service provider” to then “**download some... of that [received] software** to the **service platform server 22**” [*distributed applications for installation in the programmable network device and programmable cloud device*], Vasell does not explicitly disclose a **centralized application repository** storing the **service applications** developed by each of the “**different** service providers”, “**third party** developers” and “nonspecialists” in Vasell’s “open” development environment. *Id.*; EX1004, 7:14-18, 8:29-36, 8:47-9:18; EX1003, ¶517.

As discussed in the Motivation to Combine section, POSITAs would be motivated to modify Vasell’s teachings with Alves’ teachings of having a cloud provider give Vasell’s **service gateway system management services** access to its **application repository**, as were well-known in the art (*see* State of the Art §V.F.3.a-§V.F.3.b), in which “validate[d]” (already-tested and certified as successfully tested) **service applications** (and upgrades) would be stored, and from which Vasell’s **management services** could, as needed, download such **applications** (and

upgrades) for installation in the **service gateway system servers** [*application repository storing distributed applications which have been tested for installation in the programmable network device and programmable cloud device*] as mere application of a known solution to a known **distributed system** ready for improvement to achieve known and predictable results. *Id.*; §VII.A.5 (EX1008, 263, 266-267, FIG. 10.12, 126-130); EX1003, ¶¶518-523.

[1.8] an application management portal coupled to the programmable network device, programmable cloud device and the application repository, wherein the application management portal manages usage of the distributed applications on the programmable network device and programmable cloud device.

Vasell discloses this limitation. Vasell discloses that the **service gateway system** is “**remotely managed**”, for example, via “**management system service**” of “system services layer 110”, “implemented [as]... **boxlets**[] executing within cells”, with “**an external interface** through which the **service applications**... [are] downloaded, installed, removed, executed, and controlled”, which are **management operations** that Vasell details as being performed by a “**cell manager**” using a “**cell table**” [*application management portal coupled to the programmable network device and programmable cloud device*]:

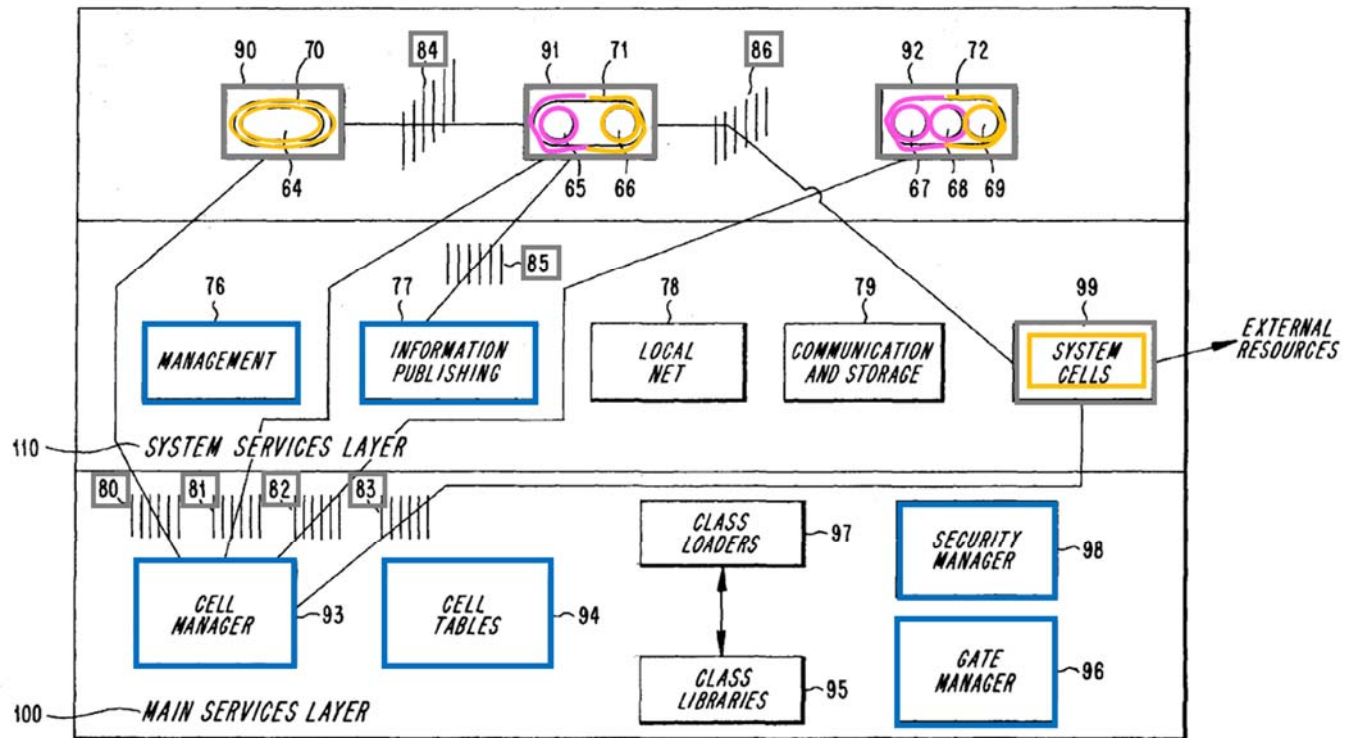


FIG. 6

EX1004, 12:11-29, 20:23-31, 24:34-37, 20:10-22:30, 15:58-17:39, 4:44-50, 5:16-26; EX1016, 6-9; EX1003, ¶¶524-526.

Per [1.7], Vasell in view of Alves discloses that Vasell’s *application management portal* is also coupled to the *application repository* from which Vasell’s *management services*, as needed, downloaded such *applications* (and upgrades) for installation in the *service gateway system servers*. EX1003, ¶527.

Vasell discloses several examples of Vasell’s *management services managing the usage of the distributed applications*. For example, Vasell discloses that the “*cell manager*”, managed via the “*management system service*” [*application management portal*], (i) “handles **resource management** so that no

single one of the [application-]cells... consumes too much of the service gateway system... resources, e.g., CPU time, memory, persistent storage...”, (ii) “manages operation of the [application-]cells... so that the right set of the cells... are running to provide tasks pertaining to the service applications... that are required at a given time” “using information stored in the cell table... which specifies the cells... that should be running on the service gateway system”, and (iii) “[a]s part of routine maintenance or in response to an error event,... stops old inactive cells... that are no longer listed in the cell table... and starts any newcomers which are listed in the cell table... but are not active or running” [manages usage of the distributed applications on the programmable network device and programmable cloud device]. EX1004, 20:45-49, 12:54-63, 13:12-27, 14:15-17, 20:36-43, 21:56-63 see also EX1004, 9:20-41 (discussing “diagnostic software which monitors the various components... needed to implement the service software”), 3:2-5, 12:27-29, 20:23-31; EX1003, ¶¶528-530, 536. POSITAs would have understood that Vasell’s disclosed structure to perform this function is the PO-indicated structure in the ’095 Patent (§V.C) or equivalent thereof. *Id.*; EX1003, ¶¶531-535.

Claim 2

The system of claim 1, wherein each of the programmable network device and programmable cloud device include an access network interface unit to send and receive

communications and a processor with a memory associated with the network interface unit.

Vasell discloses this limitation. Vasell discloses (1) “**service platform server 22**” [programmable network device] includes “access technology” to send and receive communications in the LAN and over the “Internet”, and (2) “**network operator server**” and “**service provider equipment**” include “access technology” to communicate over, and be “connected to”, “...Internet 26 or other network technology... [e.g.,] wireless”) [each of the programmable network device and programmable cloud device include an access network interface unit to send and receive communications]. EX1004, 5:64-6:4, 6:65-7:14; EX1003, ¶¶537-538.

Vasell discloses the “**cell manager**” monitors resource usage, including “CPU time, memory”, on each of these **servers** by the **service applications**, and identifies “distributed **processing** techniques... [are] implemented using **software** operating on two or more of the servers or **processors** [of Annotated FIG. 2]” [a processor with a memory associated with the network interface unit]. EX1004, 20:45-49, 14:15-17, 20:55-58, 22:49-53, 8:53-58, 9:13-19; EX1003, ¶¶539-540.

Claim 6

The system of claim 1, wherein the programmable network device and programmable cloud device further comprise: a virtual fabric which provides a secure communication layer for at the least one of the plurality of first network applications and the at least one of the plurality of second network applications.

Vasell alone, or in view of Alves, discloses this limitation. Per [1.3], Vasell alone or in view of Alves, discloses forming “distributed” **service applications** by their respective underlying components being in **secure** communication with each other.

For at least the reasons explained in [1.3], although Vasell uses different words than the '095 Patent, POSITAs understood Vasell, alone or in view of Alves, discloses, with the same, or an equivalent, structure as the PO-identified '095 Patent structure (§V.C), that the **service gateway system servers** have a *virtual fabric which provides a secure communication layer* over which the respective underlying Java components, of Vasell’s distributed **service applications**, securely and “transparent[ly]” communicate with each other using well-known “data encryption” protocols, managed “gates” enforcing cell-specific “access rules”, and the respective virtualized, sandboxed, and “higher level of abstraction” environment of a JVM running on the respective OS of the **server** on which they are hosted. EX1003, ¶¶542-549.

Claim 7

The system of claim 1, further comprising: a second programmable network device which includes the at least one of the plurality of first network applications which is in the first programmable network device.

Vasell discloses this limitation. Vasell discloses “the **service gateway system** is accessed by and shared with multiple service providers and service

gateway users”, and the “service gateway network may potentially range from several hundred thousand to millions of service gateway units.” EX1004, 4:51-54, 5:16-20. POSITAs understood Vasell discloses, including based on the examples of “service providers”, and of “connectivity based services” being provided to “various end users”, different **service platform servers** running the same “distributed” **application** “**portion**”/**component** thereon so that the same connectivity-based service is provided to different end users via the **service gateway system**. *Id.*; EX1004, 6:45-49, 3:52-57, 10:17-38, FIG. 3, 11:47-58; EX1003, ¶¶550-552.

Claim 8

The system of claim 1, wherein the programmable network device and programmable cloud device further comprise: a distributed resource service (DRS) which controls access to a plurality of resources by the plurality of first network applications and the plurality of the second network applications upon receiving instructions from the application management portal.

Vasell alone, or in view of Rellermeyer, discloses this limitation. Per [1.8], Vasell discloses (i) “**cell manager**” (itself a “**boxlet** ... executing within [a] cell”), managed via the “**management system service**” [*application management portal*], “handles **resource management** so that no single one of the [application-]cells... consumes too much of the **service gateway system**... resources, e.g., **CPU time, memory, persistent storage**...”, enforces [application-]cell-specific resource

“quota[s]”, and where this **service gateway system** “resource **management**” is facilitated by “**diagnostic software**” to “monitor”, “maintain and control” the “distributed” **service applications**. EX1003, ¶¶553-554.

Based on Vasell’s disclosures explained in [1.8], [1.3]-, POSITAs understood Vasell discloses that “access rules” (**administrator-defined** for each application’s “cell” (which “represent[s] the resources available to” the particular application) and for application cell “groups”) are controlled and enforced using, in part, “inter-cell communication... controlled interfaces” called “gates” (**managed** by “**gate manager**”), including to “limit[], monitor[], and control[]” the “resources (e.g., CPU, memory, persistent storage...)” that are respectively available, at any given point in time, to each of the underlying distributed Java components, respectively hosted in JVMs running on the OSs of the **service gateway system servers**, that form each distributed **service application**. EX1004, 12:19-29, 13:28-36, 14:2-7, Annotated FIG. 6; EX1016, 8; EX1003, ¶¶555-556.

As an example of the “resource **management**” function handled by Vasell’s “**management system service**”, *upon receiving instructions* from an administrator (e.g., upon receiving a network operator selection (and/or parameters establishing “a predetermined scheme of maximum allowed or minimum required resources, for a given situation”)) via the “**external interface** through which the **service applications** ... may be ... executed,... controlled” [*from the application*

management portal]), controlling access to **service gateway system** resources by the underlying Java components that form at least two of the distributed **service applications** (see [1.3]), Vasell discloses implementing “resource tradeoffs” for two or more “very resource-intensive” distributed **service applications** (e.g., video telephone **service application**, and 3-D interactive **video game**) in the **service gateway system** “when one of the resource intensive **service applications**... is being used and someone seeks to start another.” EX1004, 12:51-54, 13:12-27, 12:27-29; EX1016, 4, Figure 2, 7-9; EX1003, ¶¶558-559.

POSITAs understood that, as part of implementing “resource tradeoffs” for two or more “very resource-intensive” distributed **service applications** in the **service gateway system**, respective instructions would be provided from Vasell’s “**management system service**” to respective **management software** (e.g., “**service boxlet[s]**”, “**diagnostic software**”) **distributed** between the **system servers**. EX1003, ¶560.

Thus, POSITAs understood Vasell discloses this limitation. EX1003, ¶561.

To the extent PO asserts this claim requires monitoring and control functionality for load (resource usage) expansion or contraction across multiple servers in Vasell’s **service gateway system** and that Vasell does not explicitly disclose this, POSITAs would have been motivated to modify Vasell’s teachings with Rellermeyer’s teachings of providing load balancing and elastic scalability

“using features already provided by OSGi” specifications for the reasons discussed in the Motivation to Combine section. §VII.A.5 (EX1011, 2-5); EX1003, ¶¶562-565.

Claim 10

The system of claim 1, wherein the programmable network device and programmable cloud device further comprise: a distributed resource service (DRS) which is capable of providing at least one service from a group consisting of: exposing application programming interfaces (APIs) to other applications; configuring and managing platform resources; policy enforcement and authorization of applications access to platform resource and other application's APIs; and policy conflict resolution.

Vasell discloses this limitation. Per claim 8, Vasell discloses a “cell manager”, managed via the “management system service”, controlling respective, distributed management software, to “handle[] resource management” “so that no single one of the [application-]cells... consumes too much of the service gateway system... resources” [wherein the programmable network device and programmable cloud device further comprise a distributed resource service (DRS)]. EX1003, ¶567.

Vasell also discloses that the functionality provided by this distributed service includes enforcing [application-]cell-specific resource “quota[s]”, and “resource requirements... be[ing] dynamically allocated based upon a predetermined scheme of maximum allowed or minimum required resources,

for a given situation” [*distributed resource service (DRS) which is capable of providing at least one service from a group consisting of... configuring and managing platform resources*]. *Id.*; EX1004, 12:51-63, 13:12-27, 14:2-7; EX1003, ¶¶568-569.

Claim 11

The system of claim 1, wherein the programmable network device and programmable cloud device further comprise: a distributed content service wherein contents generated by the at least one of the plurality of first network applications may be shared with the at least one of a plurality of second network applications.

Vasell discloses this limitation. Specifically, POSITAs understood that part of the secure communications between the distributed **service application** components (see [1.3]) would include generated content, *e.g.*, for implementing a “freight management service”, “**client software package**... on the **service platform server**... provide[s] **freight company FMS #1** with information regarding the current location of the truck and its anticipated arrival time of delivery”

[*distributed content service wherein contents generated by the at least one of the plurality of first network applications may be shared with the at least one of a plurality of second network applications*]. EX1004, Annotated FIG. 5, 11:13-29, 8:53-9:3, 9:13-24, Annotated FIG. 2; 19:5-41 (“gates” and “proxies” used in the **service gateway system** to share “arguments and return values”, “nonserialized

objects” and “input/output data streams” between respective service-application-specific-cells); EX1003, ¶¶570-573.

Claim 12

The system of claim 1, wherein the programmable network device and programmable cloud device further comprise: a distributed notification service wherein notice of a particular event is sent from at least one of the plurality of first network applications to at least one of the plurality of second network applications.

Vasell discloses this limitation. Specifically, Vasell discloses controlled implementation of “listeners” in the service gateway system, using “proxies” to send generated “event notifications... upon the occurrence... of a specified event”, and POSITAs understood that part of the secure communications between the distributed service application components (see [1.3]) would include these “notifications... of a specified event” [notice of a particular event]. EX1004, 20:22-23, 19:21-38, 12:67-13:12; EX1003, ¶¶574-577.

Claim 13

The system of claim 10, wherein the DRS further includes a load controller; and wherein the load controller is adapted to monitor loads on at least one of the plurality of first network applications and at least one of the plurality of second network applications and effect change in accordance with thresholds received from the application management portal.

Vasell alone, or in view of Rellermeyer discloses this limitation. Per claims 8 and 10, Vasell discloses a “cell manager”, managed via the “management system

service” having “an external interface”, controlling and providing instructions to respective, distributed management software, to (i) enforce [application-]cell-specific resource “quota[s]”, (ii) “handle[] resource management so that no single one of the [application-]cells... consumes too much of the service gateway system... resources”, and (iii) “monitor”, “maintain and control” the distributed service applications implementing particular services in the service gateway system [the DRS further includes a load controller adapted to monitor loads on at least one of the plurality of first network applications and at least one of the plurality of second network applications]. EX1003, ¶¶578-579.

Vasell also discloses that the “resource management” and “monitor[ing]” functionality provided by this distributed resource service includes “resource requirements... be[ing] dynamically allocated based upon a predetermined scheme of maximum allowed or minimum required resources, for a given situation” [the DRS further includes a load controller adapted to ... effect change in accordance with thresholds received from the application management portal] so that “different ones of the service applications” do not “inadvertently interact in such a way that would cause them to... detrimentally affect each other”, and to “allow[] for resource tradeoffs between the service applications...”, which is the PO-indicated “controller executing... a... ‘load monitoring’ application” in the ’095 Patent (§V.C) or equivalent thereof. *Id.*; EX1004, 12:51-63, 13:12-27; see

also claims 11, 12 (identifying Vasell’s disclosure of *distributed content*, and *distributed notification, services*); EX1003, ¶¶580-588.

To the extent PO asserts this claim requires monitoring and control functionality for load (resource usage) expansion or contraction across multiple servers in Vasell’s *service gateway system* and that Vasell does not explicitly disclose this, POSITAs would have been motivated to modify Vasell’s teachings with Rellermeier’s teachings of load balancing and elastic scalability, which discloses the “controller executing... a ‘cloud breathing’ [**and**] ‘load monitoring’ application” indicated in the ’095 Patent (§V.C) or equivalent thereof, as discussed in the Motivation to Combine section. §VII.A.5 (EX1011, 2-5); EX1003, ¶¶589-593.

Claim 15

Claim 15 is a system claim corresponding to the claim 1 elements, with exceptions noted (by underlining) below, and the Claim 15 limitations are invalid for the reasons discussed regarding Claim 1 and additional reasons discussed below.

[15.0] A system for processing data packets, the system comprising:

Vasell discloses this limitation per [1.0].

[15.1] a programmable network device processing data flows hosting at least one of a plurality of first network applications;

Vasell discloses this limitation per [1.0]-[1.1]. Specifically, Vasell discloses examples of the applications' components on service platform server [programmable network device] implementing telephony services such as IP telephony or video telephony, which POSITAs understood discloses processing data flows. EX1004, 9:46-68, 6:34-39, 12:54-57, 13:15:18, 1:54-55, 5:52-56; EX1003, ¶596.

[15.2] a programmable cloud device having a plurality of virtual machines wherein at least one of the virtual machines hosts at least one of a plurality of second network applications;

Vasell alone, or in view of Alves, discloses this limitation. Specifically, Vasell discloses that remote (cloud) servers in the service gateway system may utilize “multiple JVMs” running on the respective servers’ OSs, and may include multiple “network operators using the network operator servers” and many “different service providers”, each with respective service provider equipment. EX1004, 18:23-30, 13:28-29, 13:50-51, 14:1-2, 6:49-65, 2:60-64, 9:41-46, 11:7-10; EX1016, 4-6, 7, 9; EX1003, ¶¶598-600.

To the extent PO asserts Vasell does not explicitly disclose each disclosed JVM is run on top of a host OS running on each “remote” (cloud) server’s hardware, POSITAs would have been motivated to do so because, as discussed in the State of the Art §V.F.1, §V.F.3.a, §V.F.3.b, this was one of many well-documented technical implementation details in OSGi literature and would provide

the readily apparent and predictable benefit of implementing Vasell's **distributed** processing in its standardized and conventional way. §VII.A.5; §V.F.3.a (EX1008, xiv, 2, 6-7, 16, 250, 261-263; EX1013, 3-4, 8, 10, 11; EX1029, 1-2; EX1030, FIG. 2; EX1010, FIG. 2); §V.F.1 (EX1031, 569-570); §V.F.3.b (EX1052, xxvi, 10-12, 19-20, 51-52, 94-95, 100-103); EX1003, ¶¶601-602.

To the extent PO asserts (i) the claims require that a single **remote (cloud) server** have the “plurality of virtual machines”, and that one JVM on this single **server** host the “at least one of the plurality of second network applications”, and (ii) Vasell does not explicitly disclose this limitation, as discussed in the State of the Art §V.F.1, §V.F.3.b, doing so was a well-known design choice to POSITAs, with well-known benefits (*e.g.*, “prioritiz[ing] application execution on OS level” for “an application requir[ing] higher isolation”; “[r]esources... are assigned to each VM when... created, yet often can be dynamically varied at run time”, etc.) and trade-offs (*e.g.*, “performance... decrease[s] if many VMs are attached”) associated with its implementation. §V.F.1 (EX1031, 569-570; EX1036, 278-279); §V.F.3.b (EX1030, [0033]-[0037]; EX1052, 69-70, 82; EX1012, 19-22, FIG. 1; EX1043, 124-125; EX1039, 1-3); EX1003, ¶¶603-605.

POSITAs would recognize that Vasell does not go into detail on implementing its **service gateway system** solution by selecting the design choice of at least one of its **remote (cloud) servers** having a plurality of VMs (“multiple

JVMs”), over, for example, the design choice of each of its **remote servers** having a respective VM, because, as acknowledged by the ’095 Patent (which also does not provide such details, *see* EX1001, 17:12-25, FIG. 8), such selection, with well-known advantages and trade-offs, was well within the knowledge, skill and creativity of POSITAs. *Id.*; EX1003, ¶¶604-605.

[15.3] wherein the at least one of the plurality of first and the at least one of the plurality of second network applications are in secure communication to form a distributed application;

Vasell alone, or in view of Alves, discloses this limitation per [1.3].

[15.4] the programmable network device and programmable cloud device are each powered by a sandboxing operationu system which facilitates deployment of the plurality of first and second network applications

Vasell alone, or in view of Alves, discloses this limitation per [1.4].

[15.5] and [the sandboxing operationu system] facilitates upgrade of the plurality of first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device;

Vasell alone, or in view of Alves, discloses this limitation per [1.5].

[15.6] wherein the programmable network device verifies authenticity of the upgrades to the plurality of first network applications and the programmable cloud device verifies the authenticity of the upgrades to the plurality of second network applications and wherein the verification is based on unique security keys associated with each of the plurality of first and second network applications;

Vasell alone, or in view of Hall, discloses this limitation per [1.6].

[15.7] an application repository storing distributed applications which have been tested for installation in the programmable network device and programmable cloud device; and

Vasell in view of Alves, discloses this limitation per [1.7].

[15.8] an application management portal coupled to the programmable network device, programmable cloud device and the application repository, wherein the application management portal manages provisioning, usage and de-provisioning of the distributed applications on the programmable network device and programmable cloud device.

Vasell discloses this limitation. Per [1.8], Vasell discloses its **management components** [application management portal], **managing service application** upgrades, and **manages** usage of the **distributed applications**, on the respective **servers** in the **service gateway system**.

Vasell further discloses these **management components** “facilitate **management** of the **service applications**... lifetimes”, “provides an **external interface** through which the **service applications**... may be **downloaded, installed, removed...**”, “is responsible for **starting or activating** [application-]cells... as well as and **stopping and destroying** [application-]cells...”, “stor[e] the state of the **service applications**... to boot the **service applications**... of the cells... in any order”, and “manag[e] the **creation** and **destruction** of the [application-]cells”, on the respective **service gateway system servers** [bold emphasis indicating *further*

manages provisioning... and de-provisioning of the distributed applications].

EX1004, 22:35-38, 20:32-34, 20:40-43, 20:49-53, 21:13-26; Annotated FIG. 6; EX1016, 7-8; EX1003, ¶¶611-615.

VIII. Discretionary Denial Would be Inappropriate

A. Discretionary Denial under the *Fintiv* Factors is Inappropriate

Under *Fintiv*, the Board takes a holistic view of whether efficiency, fairness, and the merits support the exercise of authority to deny institution in view of an earlier trial date in the parallel proceeding. *Apple Inc. v. Fintiv, Inc.*, Case IPR2020-00019, Paper 11, 6 (March 20, 2020).

According to the latest guidance from the Director:

The Board should first assess *Fintiv* factors 1–5; if that analysis supports discretionary denial, the Board should engage the compelling merits question. If the Board reaches the compelling merits analysis and finds compelling merits, it shall provide reasoning to explain its determination.

CommScope Techs. LLC, v. Dali Wireless Inc., IPR2022-01242, Paper 23, 6 (February 27, 2023).

Factor 1 is neutral because no request for stay has been filed in the Related Litigation.

Factor 2 favors institution. No trial date has been set in the Related Litigation, which was filed February 29, 2024 in W.D. Texas where, according to

the most recent statistics for civil actions², the median time from filing to trial is 33.1 months; accordingly, trial would be expected in late-November 2026. A Final Written Decision is expected in the present matter in September 2026, months before an expected trial in the Related Litigation.

Factor 3 is neutral. Fact discovery and claim construction are ongoing and, at the time of an expected institution decision, according to the operative Scheduling Order (EX1073), fact discovery will have closed and expert discovery will be ongoing.

Under factor 4, there will be no overlap of prior art issues as the Related Litigation does not involve claims 6, 8, and 10-13 challenged herein, and Petitioner stipulates that, if trial is instituted, it will not pursue the same grounds in the litigation. *See Sand Revolution II, LLC v. Cont'l Intermodal Group-Trucking LLC*, IPR2019- 01393, Paper 24, 12 (PTAB June 16, 2020). Consequently, this factor favors institution.

Factor 5 is neutral because, although Petitioner is a litigation defendant, this is “far from an unusual circumstance” for petitioners in IPR proceedings. *Sand Revolution*, Paper 24, 12-13.

² https://www.uscourts.gov/sites/default/files/2024-12/fcms_na_distprofile0930.2024.pdf.

To the extent that the Board finds that factors 1-5 favor discretionary denial, under factor 6, other circumstances, including compelling merits, weigh in favor of institution. For example, Vasell is a parallel disclosure to the '095 Patent, meaning that it is directed at a substantially similar problem and proposes a substantially similar solution as the '095 Patent, and discloses the same architecture and the same concepts disclosed in the '095 Patent, but uses some different words and leaves certain implementation details to design choices of POSITAs. Vasell's architecture and methods were foundational concepts (including implementation in a Java run-time environment) that were eventually standardized under OSGi. Technical implementation details that were intended to implement Vasell's teachings are described in each of the Ground 1 references—Alves, Hall and Rellermeyer. Accordingly, the motivation for POSITAs to modify the teachings of Vasell, to include the features described in these references for their intended purpose and to achieve predictable benefits with a reasonable expectation of success, is demonstrated by the references themselves, thus demonstrating that the Petition is particularly strong in the underlying merits. The evidence presented, if unrebutted in trial, would plainly lead to a conclusion that one or more claims are unpatentable by a preponderance of the evidence.

When viewed holistically, the efficiency, fairness, and merits support instituting review.

B. Discretionary Denial Under §325(d) is Not Appropriate

Petitioner is unaware of any authority holding *Advanced Bionics* to be satisfied where (like here) the prosecution history lacks discussion of any reference relied upon in this petition. In such situations, the Board routinely does not decline institution on §325 grounds. Moreover, the Petition's challenges are non-cumulative because the prior art presented describes the claim limitations that were added to gain allowance. *See* §V.A-§VII.A.6.

IX. Mandatory Notices Under 37 C.F.R. §42.8

A. Real Party-in-Interest (37 C.F.R. § 42.8(b)(1))

The real party-in-interest in this Petition is Microsoft Corporation.

B. Related Matters (37 C.F.R. § 42.8(b)(2))

1. Judicial Matters

As of the filing date of this Petition and to the best knowledge of Petitioner, the '095 Patent is involved in the following litigations:

Edge Networking Systems, LLC v. Microsoft Corporation, Case No. 1:24-cv-215-DAE (W.D. Tex.) (filed Feb. 29, 2024) (ongoing); and

Edge Networking Systems LLC v. Amazon.Com, Inc., et al., Case No. EDTX-2-24-cv-00887 (E.D. Tex.) (filed Nov. 1, 2024) (ongoing).

Administrative Matters:

As of the filing date of this Petition and, to the best knowledge of Petitioner, the '095 Patent has not been subject to any *inter partes* reviews, reissues, or

reexaminations. Concurrently with the filing of this IPR petition, Petitioner is filing a petition for IPR of related U.S. Pat Nos. 10,686,871 (the “’871 Patent”) and 10,695,823 (the “’823 Patent”).

2. Related Patents

As of the filing date of this Petition and to the best knowledge of Petitioner, the following patent applications and patents are related to the ’095 Patent:

Patent/Publication Number	Application Number	Filing Date	Issue Date	Exhibit No.
9,843,624	14/295,331	6/4/2014	12/12/2017	EX1005
10,686,871	15/836,824	12/9/2017	6/16/2020	EX1006
11,695,823	17/142,983	1/6/2021	7/4/2023	EX1007
12,126,673	18/217,332	6/30/2023	10/22/2024	
12,113,850	18/600,742	3/10/2024	10/8/2024	
12,126,674	18/600,747	3/10/2024	10/8/2024	
12,113,851	18/664,279	5/14/2024	10/22/2024	
	18/824,854	9/4/2024		

C. Lead/Back-up Counsel (37 C.F.R. § 42.8(b)(3)):

Lead Counsel:

Christopher J. Tyson, USPTO Reg. No. 63,850

DUANE MORRIS LLP, 901 New York Avenue N.W., Suite 700 East,
Washington, D.C. 20001

P: (202) 776-7851; F: (202) 478-2620; CJTyson@duanemorris.com

Back-Up Counsel:

John M. Baird, USPTO Reg. No. 57,585

DUANE MORRIS LLP, 901 New York Avenue N.W., Suite 700 East,
Washington, D.C. 20001

P: (202) 776-7819; F: (202) 478-2620; JMBaird@duanemorris.com
Glenn D. Richeson, USPTO Reg. No. 73,480 DUANE MORRIS LLP, 1075 Peachtree Street NE, Suite 1700 Atlanta, GA 30309 P: (404) 253-6993; F: (404) 759-2703; GDRicheson@duanemorris.com
Patrick D. McPherson, USPTO Reg. No. 46,255 DUANE MORRIS LLP, 901 New York Avenue N.W., Suite 700 East, Washington, D.C. 20001 P: (202) 776-5214; F: (202) 478-0826; PDMcPherson@duanemorris.com

D. Notice of Service Information (37 C.F.R. § 42.8(b)(4)):

Please direct all correspondence to lead and back-up counsel at the above addresses. Petitioner consents to electronic service at the email addresses above.

X. CONCLUSION

Petitioner requests the Board institute IPR and cancel the Challenged Claims.

Respectfully submitted,

DUANE MORRIS LLP

/Christopher J. Tyson/
Christopher J. Tyson, Reg. No. 63,850
901 New York Avenue, N.W.
Suite 700 East
Washington, D.C. 20001-4795
P: (202) 776-7851
F: (202) 478-2620
CJTyson@duanemorris.com

ATTORNEY FOR PETITIONER

Dated: February 17, 2025

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. §42.24 *et seq.*, the undersigned certifies that this document complies with the type-volume limitations. This document contains 13,982 words as calculated by the “Word Count” feature of Microsoft Word 365, the word processing program used to create it.

Dated: February 17, 2025

/Christopher J. Tyson/

Christopher J. Tyson, Reg. No. 63,850
901 New York Avenue, N.W.
Suite 700 East
Washington, D.C. 20001-4795
P: (202) 776-7851
F: (202) 478-2620
CJTyson@duanemorris.com

ATTORNEY FOR PETITIONER

CERTIFICATION OF SERVICE ON PATENT OWNER

Pursuant to 37 C.F.R. §§ 42.6(e), 42.8(b)(4) and 42.105, the undersigned certifies that on February 17, 2025, a complete and entire copy of this Petition for *Inter Partes* Review of U.S. Patent No. 10,893,095 and all supporting exhibits were served on the Patent Owner by serving, via email, the counsel of record for the Patent Owner at the USPTO for the '095 Patent (Christopher Horgan, ROARK IP, chris.horgan@roarkip.com) and by serving, via email, counsel for the Patent Owner in the Related Litigation (Hecht Partners LLP, David L. Hecht (dhecht@hechtpartners.com); Maxim Price (mprice@hechtpartners.com); Yi Wen Wu (wwu@hechtpartners.com); Tremayne M. Norris (tnorris@hechtpartners.com)). The undersigned also certifies that on February 18, 2025, a complete and entire copy of this Petition for *Inter Partes* Review of U.S. Patent No. 10,893,095 and all supporting exhibits, will be served via Federal Express, postage prepaid, on the Patent Owner by serving the correspondence address of record for the '095 Patent:

ROARK IP
1438 Dahlia Loop
San Jose, CA 95125

/Christopher J. Tyson/
Christopher J. Tyson, Reg. No. 63,850
901 New York Avenue, N.W.
Suite 700 East
Washington, D.C. 20001-4795

Petition for *Inter Partes* Review of U. S. Patent No. 10,893,095

P: (202) 776-7851

F: (202) 478-2620

CJTyson@duanemorris.com

ATTORNEY FOR PETITIONER