

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION

Petitioner,

v.

EDGE NETWORKING SYSTEMS, LLC,

Patent Owner.

Case No. IPR2025-00617
U.S. Patent No. 10,893,095

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

PATENT OWNER’S EXHIBIT LIST	v
I. INTRODUCTION.....	1
II. THE SPECIFICATION OF THE ‘095 PATENT	4
A. Background	4
B. Overview Of Distributed Software Defined Network With Sandboxing Operating System.....	4
III. PROSECUTION HISTORY OF ‘095 PATENT	7
A. Prosecution History of ‘624 Patent.....	10
1. Rejection Of Claims Submitted With Request For Continued Examination	10
2. First Examiner Interview and Response To Rejection	18
3. Second Examiner Interview.....	21
4. Examiner’s Amendment	21
5. Prosecution History of ‘095 Patent.....	26
IV. CLAIMS OF THE ‘095 PATENT	29
V. CLAIM CONSTRUCTION	32
A. In The District Court Petitioner Maintains That A “Sandboxing Operating System” Must Have A Kernel But In This Proceeding Petitioner Ignores Any Requirement That The “Sandboxing Operating System” Have A Kernel.....	33
B. Petitioner’s Indefiniteness Positions In The District Court Cannot Be Reconciled With Petitioner’s Interpretation Of The Claims Of The ‘095 Patent In This IPR.....	34
C. To The Extent The Petition Advances Any Construction For The Substantially No Interruption Claim Terms, The Petition’s Construction Requires Structure That Permits One Instance Of The Application To Remain Running During The Upgrade	36
D. To The Extent The Petition Advances Any Construction For The Usage Management Claim Terms, The Petition’s Construction Requires Structure That Uses Information In A Secure Manifest File	36

E.	The Petition Fails To Articulate How The Challenged Claims Are To Be Construed	37
VI.	APPLICABLE LEGAL STANDARDS	38
VII.	A PERSON HAVING ORDINARY SKILL IN THE ART	38
VIII.	THE VIRTUALIZED APPLICATION RUNTIME ENVIRONMENT IN SHATZKAMER CONSIDERED DURING PROSECUTION.....	39
A.	The Virtualized Runtime Environment Of Shatzkamer	39
B.	The Petition Explains That A Virtualized Runtime Environment Is Distinct From The Operating System On Which Virtualized Runtime Environment Sits	40
C.	The Petition Conflates Isolation In General With Isolation At The Operating System Level As Recited In The Claims Of The ‘095 Patent	42
IX.	PRIOR ART RELIED ON IN PETITION.....	43
A.	Vasell (Ex. 1004)	43
1.	The Virtual Runtime Environment In Vasell Is Cumulative Of That Disclosed In Shatzkamer	43
2.	Vasell Discloses A Process For Updating A Configuration During Which Operation Of The Service Platform Server Stops And Starts.....	44
B.	Alves (Ex. 1008)	45
C.	Hall (Ex. 1009).....	45
D.	Rellermeyer (Ex. 1011).....	45
X.	GROUND 1 FAILS TO RENDER ANY OF THE CHALLENGED CLAIMS UNPATENTABLE BECAUSE THE PETITION HAS FAILED TO DEMONSTRATE THAT THE RELIED ON REFERENCES DISCLOSE A SANDBOXING OPERATING SYSTEM	45
A.	The Petition Has Failed To Demonstrate That Vasell Discloses A Sandboxing Operating System.....	45
B.	Petitioner’s Conclusory Assertion That The Art Presented In The Petition Describes The Claim Limitations Added To Gain Allowance Is False	47

XI.	GROUND 1 FAILS TO RENDER ANY OF THE CHALLENGED CLAIMS UNPATENTABLE BECAUSE THE PETITION HAS FAILED TO EXPLAIN HOW VASELL MEETS THE SUBSTANTIALLY NO INTERRUPTION CLAIM TERMS.....	48
XII.	GROUND 1 FAILS TO RENDER ANY OF THE CHALLENGED CLAIMS UNPATENTABLE BECAUSE THE PETITION HAS FAILED TO EXPLAIN HOW VASELL MEETS THE USAGE MANAGEMENT CLAIM TERMS.....	51
XIII.	CONCLUSION	52

PATENT OWNER'S EXHIBIT LIST

Ex. 2001	US Patent Publ. No. 2014/0317293 to Shatzkamer
Ex. 2002	McKeown et al, "Open Flow: Enabling Innovation in Campus Networks" March 14, 2008
Ex. 2003	US Patent Publ. No. 2013/0160147 to Draluk
Ex. 2004	US Patent Publ. No. 2004/0237081 to Homiller
Ex. 2005	<i>Ruckus Wireless, Inc. v. Hera Wireless SA et al.</i> , IPR2018-01739, Petition for Inter Partes Review, Paper 1
Ex. 2006	US Patent No. 10,496,387
Ex. 2007	Office Action dated April 3, 2019 issued during prosecution of US Patent No. 10,496,387 together with Notice of References cited accompanying same

I. INTRODUCTION

The Petition challenges claims 1-2, 6-8, 10-13 and 15 (“Challenged Claims”) of U.S. Patent No. 10,893,095 (“ ‘095 Patent,” Ex. 1001) under a single ground. Claims 1 and 15 are the only Challenged Claims that are independent. The Board should deny the institution for at least the following reasons.

First, each of the Challenged Claims requires a programmable network device and programmable cloud device that are *each powered by a sandboxing operating system*. The recited “sandboxing operating system” was critical to the allowance of the ‘095 Patent. More specifically, sandboxing at the *operating system level* served to distinguish the claims from art in front of the Examiner like Shatzkamer that disclosed only *application-level isolation*. With respect to this limitation, the Petition has failed to show that the Vasell reference relied on in the Petition provides anything beyond the *application-level isolation* disclosed in the Shatzkamer reference applied by the Examiner and correctly found wanting.

Second, the Petition has failed to articulate how it is interpreting (i) managing/facilitating “upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device” as recited in claims 1 and 15 (referred to for shorthand as the “Substantially No Interruption Claim Terms”), or (ii) the requirement in claim 1 that “the application management portal manages usage of the distributed

applications” and the requirement in claim 15 that “the application management portal manages provisioning, usage and deprovisioning the distributed applications” (referred to collectively for shorthand as the “Usage Management Claim Terms”). Rather than specifying how the Petition is construing these limitations as required by the rules, the Petition maintains that “the Ground renders the Challenged Claim unpatentable under either party’s construction, and any reasonable construction.” Petition, 10-11. However, in the district court, the Petitioner maintains that the Substantially No Interruption Claim Terms and the Usage Management Claim Terms have no construction because they are indefinite. Ex. 1070, 12-14, 15-16 and Ex. 1072, 15-17, 18-19. In the Petition itself, Petitioner advances no construction of its own for these limitations, instead providing only “specific portions of the specification that describe the structure” of the claims “to the extent Patent Owner (“PO”) argues or the Board finds they are governed by means-plus-function.” Petition, 10-11, 13 (emphasis added). This approach fails to satisfy the requirement that the Petition (rather than the Board or Patent Owner) articulate in the first instance how the claim language is being constructed.

Third, in the district court the Petitioner has stated that Patent Owner concedes that the Substantially No Interruption Claim Terms are governed by a means plus function analysis under 35 USC §112(f). Ex. 1072, 16. Based on this so-called concession, the Petition identifies the following structure for the Substantially No

Interruption Claim Terms: “[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” Petition, 12 (emphasis added). Even if the identification of such structure satisfied Petitioner’s burden to articulate how the Substantially No Interruption Claim Terms were to be construed (to be clear, it does not), the Petition is still deficient because the Petition nowhere explains how the identified structure corresponding to the Substantially No Interruption Claim Terms is met by the relied on references. For example, the Petition nowhere identifies structure that permits one instance of the application to remain running during the upgrade in the relied on references.

Fourth, in the district court the Petitioner has stated that Patent Owner concedes that the Usage Management Claims Terms are governed by a means plus function analysis under 35 USC §112(f). Ex. 1072, 18. Based on this so-called concession, the Petition identifies the following structure for the Usage Management Claims Terms: “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.” Petition, 13 (emphasis added). Even if the identification of such structure satisfied Petitioner’s burden to articulate how the Usage Management Claims Terms were to be construed (to be

clear, it does not), the Petition is still deficient because the Petition nowhere explains how the identified structure corresponding to the Usage Management Claim Terms is met by the relied on references. For example, the Petition nowhere identifies a “*manifest file*” in the relied on references.

For at least these reasons, and as explained more fully below, institution should be denied.

II. THE SPECIFICATION OF THE ‘095 PATENT

A. Background

The ‘095 Patent relates to cloud computing and dates back to 2013. At that time, “[e]xisting network solutions [were] built on proprietary hardware and software” that resulted in network operators being “unable to add new and customized features or capabilities.” Ex. 1001, 1:24-28. In these systems, network operators who wanted to “add new desired features” were required to make “requests to their infrastructure vendors or pursu[e] standardization processes.” *Id.*, 1:29-31. These approaches were “time consuming and resource intensive.” *Id.*, 1:31-32.

B. Overview Of Distributed Software Defined Network With Sandboxing Operating System

The ‘095 Patent disclosed a “Distributed Software Defined Network (dSDN)” which “is an end-to-end architecture that enables secure and flexible programmability across a network with full lifecycle management of services and

applications.” *Id.*, 7:57-59. “In the dSDN architecture, the network features are virtualized and may be distributed across separate network elements that cooperate together to create an advanced, programmable, and scalable network.” *Id.*, 10:2-6.

“A high-level overview of a dSDN system 300” is depicted in Figure 3 of the ‘095 Patent (reproduced below -- colored annotations added by Patent Owner). *Id.*, 10:8-9. “The dSDN system 300 consists of a flexible network device (fxDevice) 302 [red box], a flexible cloud platform (fxCloud) 304 [blue box], an application management portal (fxManager) 306 [orange box], and an infrastructure application market place (fxStore) 308 [green box].” *Id.*, 10:9-13.

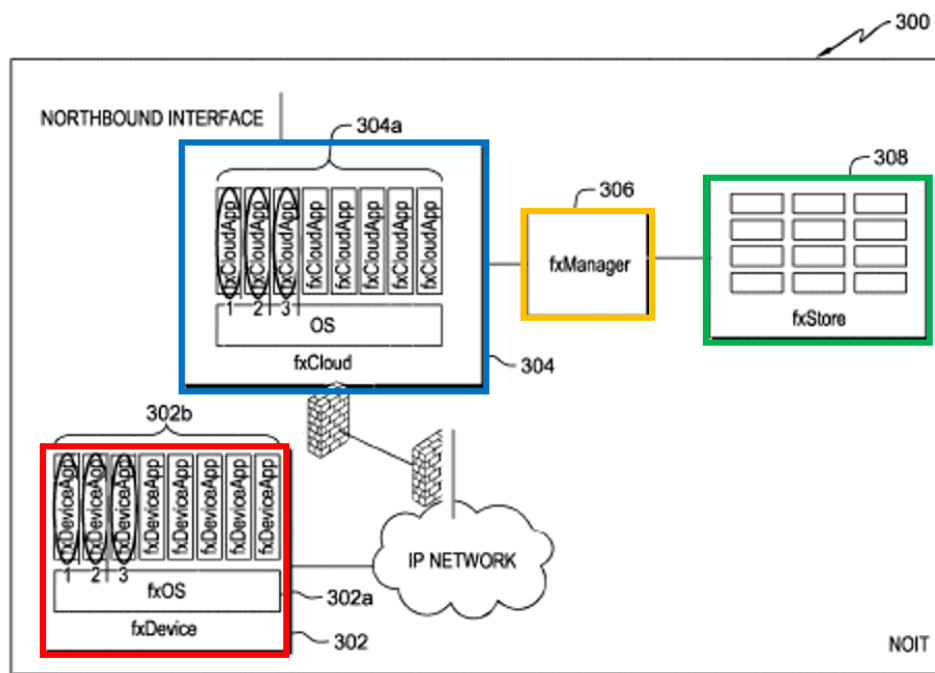


FIG. 3

Significantly, “[t]he fxDevice 302 [red box] is a network device that is powered by

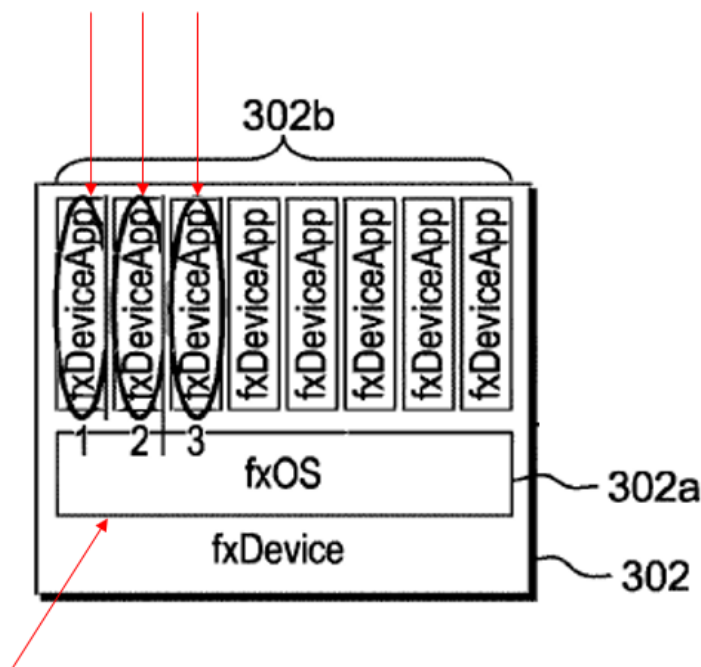
a sandboxing operating system named flexible operating system (fxOS) 302a.” *Id.*,

10:9-13 (emphasis added). As explained in the ‘095 Patent:

... A high-level overview of a dSDN system 300 is depicted in FIG. 3. The fxDevice 302 is a network device that is powered by a sandboxing operating system named flexible operating system (fxOS) 302a. The sandboxing operating system ensures each application runs as a dedicated process in sure isolation from the other applications. The fxOS 302a may be built based on an existing sandboxed operating system (OS) (e.g., Android®) by extending several aspects of such an OS with routing/networking layers and supports for operating as the wireless infrastructure. The fxDevice 302 may host several independent and securely isolated applications (on top of fxOS 302a) named here fxDeviceApp 302b. *Id.*, 10:8-24 (emphasis added).

The disclosed sandboxing operating system is illustrated in connection with the fxDevice 302 shown in Figure 3. The portion of Figure 3 depicting fxDevice 302 is reproduced in part below with Patent Owner’s annotations (in red):

Applications running in
isolation from each other



Sandboxing operating system

As shown above, the “sandboxing operating system named flexible operating system (fxOS) 302a” hosts “several independent and securely isolated applications (on top of fxOS 302a).” Ex. 1001, 10:8-24. A sandboxing operating system like fxOS 302a disclosed in the ‘095 Patent provides *isolation at the operating system (OS) level*.

III. PROSECUTION HISTORY OF ‘095 PATENT

The Petition challenges the ‘095 Patent based on a combination of prior art that is, at best, cumulative of the art that the Examiner applied during prosecution of the grandparent of the ‘095 Patent, i.e., U.S. Patent No. 9,843,624 (the ‘624 Patent). Just prior to allowance, the Examiner of the ‘624 Patent maintained that the art being applied met the pending claims. Allowance of the ‘624 Patent was subsequently

based on an Examiner's amendment requiring that the programmable network device previously recited in the claims be narrowed to a programmable network device "powered by a sandboxing operating system." The claims of the '095 Patent substantially track those of the '624 Patent, and carry forward the requirement of a programmable network device "powered by a sandboxing operating system" that was critical to patentability. With respect to this particular limitation, the reference advanced in the Petition (Vasell) is cumulative of the reference that the Examiner applied and correctly found wanting, and the Petition has failed to explain how the Examiner erred during prosecution (to be clear, there was no error).

During the prosecution of the '624 Patent, the Examiner rejected the claims over a combination of multiple references which are discussed in detail below. One of those references -- U.S. Patent Publ. No. 2014/0317293 to Shatzkamer (Shatzkamer) (Ex. 2001) -- disclosed a "virtualized application runtime environment" for running network applications. However, the Examiner recognized that Shatzkamer failed to implement isolation of applications at the "operating system" level. In view of this deficiency, the Examiner introduced (by way of an Examiner's Amendment) into the claims a further requirement that the programmable network device be "powered by a sandboxing operating system." In the Examiner's reasons for allowance of the '624 Patent, the Examiner expressly found that this aspect of the claims was absent from Shatzkamer. As discussed more

fully below, with respect to the requirement that the programmable network device be “powered by a sandboxing operating system,” the reference advanced in the Petition (Vasell) is cumulative of the Shatzkamer reference that the Examiner correctly found wanting.

When multiple patents derive from the same initial application, the prosecution history regarding a claim limitation in any patent that has issued applies with equal force to subsequently issued patents that contain the same claim limitation. *Elkay Mfg. Co. v. Ebco Mfg. Co.*, 192 F.3d 973, 980 (Fed. Cir. 1999). Here, the ‘095 Patent and the ‘624 Patent derive from the same application, because the ‘095 Patent is a continuation of U.S. Patent No. 10,686,871, which is a continuation of the ‘624 Patent. Thus, the prosecution history regarding a claim limitation in the ‘624 Patent (i.e., the limitation reciting “a sandboxing operating system”) applies with equal force to the subsequently issued ‘095 Patent because, as explained below, both patents contain the same limitation.

As demonstrated in detail below, the claims of the ‘095 Patent substantially track those of the ‘624 Patent. Among other similarities, the claims of the ‘095 Patent carry forward from the ‘624 Patent the requirement of a programmable network device “powered by a sandboxing operating system.” The similarities between the claims of the ‘624 Patent and ‘095 Patent are underscored by the prosecution of the

‘095 Patent, which was limited to an obviousness-type double-patenting rejection over, *inter alia*, the ‘624 Patent.

The Petition advances a combination of prior art that is, at best, cumulative of the art that the Examiner applied during the prosecution of the ‘624 Patent. Most importantly, with respect to the limitation upon which allowance was based -- i.e., the requirement of a programmable network device “powered by a sandboxing operating system” -- the Vasell reference advanced in the Petition is cumulative of the Shatzkamer reference applied during prosecution.

A. Prosecution History of ‘624 Patent

1. Rejection Of Claims Submitted With Request For Continued Examination

The application that matured into the ‘624 Patent was filed on June 4, 2014. Ex. 1005, 2. Following two earlier rejections (Ex. 1005, 151-176, 242-273), the Applicant filed a Request for Continued Examination, which included, *inter alia*, claims 1, 7, 17 and 18 set forth below:

1. (Currently Amended) A system ~~comprising~~ for processing data packets in a mobile data network that includes a base station coupled to a core network having a plurality of network elements, said network elements comprising:

a programmable network device adapted to host a plurality of first network applications;

a programmable cloud device adapted to host a plurality of second network applications, wherein the plurality of first network applications in the programmable network device and the plurality of second network applications in the programmable cloud device are in

secure communication with each other to form distributed applications;
~~and~~

the programmable network device and programmable cloud device forming unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) that allow for ~~simultaneous~~ programming of the programmable network device and programmable cloud device independent of the programmable network device hardware and the programmable cloud device hardware;

an infrastructure application market place in communication with an application management portal coupled to the programmable network device and programmable cloud device, said infrastructure market place capable of providing tested and certified distributed applications to the application management portal for installation in the first network applications and second network applications and allow the application management portal to manage life cycles of the distributed applications; and

wherein the application management portal is further capable of live upgrading of the distributed applications with substantially no interruption to operation.

7. (Previously Presented) The system of claim 1, wherein the programmable network device is an LTE eNB and the programmable cloud device hosts an LTE core network functions in separate virtual machines from the group consisting of: MME, S-GW, and P-GW.

17. (Previously Presented) The system of claim 1, wherein the at least one of the plurality of first network applications and at least one of the plurality of second network applications are given an access level to determine which APIs can be accessed.

18. (Currently Amended) The system of claim 1, wherein the ~~at least one~~ programmable network device ~~and programmable cloud device are~~ is adapted to verify the authenticity and integrity of updates to the first network applications and the programmable cloud device is adapted to verify the authenticity and integrity of updates to the second network applications.

Ex. 1005, 303-307.

The Examiner responded to the Request for Continued Examination by rejecting all of the pending claims over prior art. With respect to pending claims 1, 7, 17 and 18 set forth above, the Examiner applied combinations of six references including Li (Ex. 1005, 371-376), the OpenFlow white paper (Ex. 2002), Shatzkamer (Ex. 2001), the NFV white paper (Ex. 1005, 377-392), Draluk (Ex. 2003) and Homiller (Ex. 2004). Ex. 1005, 321-344. Before getting into the details of how these references mapped to the claims, the Examiner stated:

In response to Applicant's patentability claim, the Examiner respectfully disagrees noting that most of the claimed features are already available in well-known industry standards such as Software Defined Networking (SDN), OpenFlow Protocol, and Network Function Virtualization (NFV). SDN, OpenFlow and NFV are already geared towards hardware-independent implementation, in-service ("hot") software upgrade, and secure communications between entities. Further, Shatzkamer et al (2014/0317293) discloses an app-store portal for providing interdependent virtualized network functions for different types of services (distributed applications), such as an Evolved Packet service [0031]. Similarly, Li et al ("Toward Software Defined Cellular Networks") discloses using SDN to simplify design and management of cellular data networks. Also, the idea of distributed management and control is not new to LTE, where the functions of a radio network controller are distributed between the radio base stations (eNodeB) and the access gateways in the core (SGW, PGW, MME, etc.). Therefore, *it is not clear what the novel and inventive features of this Application are.* Ex. 1005, 323 (emphasis in original).

Following this overview, the Examiner presented his view of how the claims mapped to the prior art.¹ With respect to claims 1, 17 and 18 (set forth above), the chart below sets forth the prior art mappings from the rejection:

<u>Elements from Claim 1</u>	<u>Examiner's Mapping To Prior Art</u>
A system for processing data packets in a mobile data network that includes a base station coupled to a core network having a plurality of network elements, said network elements comprising:	“Li discloses the method of processing data packets in a mobile data network that includes a base station coupled to a core network (<i>In LTE, the user equipment UE connects to a base station (eNodeB), which directs traffic through a serving gateway S-GW which tunnels the traffic to a packet gateway P-GW which in turn connects the traffic to the internet. ...</i>)” Ex. 1005, 327 (mapping corresponding feature in independent claim 22 to Li) and 332 (applying same analysis to claim 1).
a programmable network device adapted to host a plurality of first network applications; a programmable cloud device adapted to host a plurality of second network applications,	Li discloses that “Cellular data networks are ripe for the introduction of Software Defined Networking (SDN), where the network equipment performs basic packet-processing functions (i.e. first network applications hosted by network device) at the behest of applications running on a logically-centralized controller (i.e. second network applications hosted by cloud device, such that network equipment and the controller provide distributed service to process cellular data traffic) [page 2]; using a programmable control plane and a programmable data plane [page 5, last paragraph]. “

¹ Patent Owner does not concede that the Examiner's mappings set forth herein, or the Examiner's rationale for combining references, are correct.

	<i>Id.</i> , 328 (mapping corresponding features in independent claim 22 to Li) and 332 (applying same analysis to claim 1).
wherein the plurality of first network applications in the programmable network device and the plurality of second network applications in the programmable cloud device are in secure communication with each other to form distributed applications;	“OpenFlow discloses that the applications in the network devices are in secure communication with the applications in the cloud devices (a secure channel that connects the OpenFlow switch to the remote control process in the controller [page 2 last paragraph]).” <i>Id.</i>, 331 (mapping corresponding “secure communication” feature in claim 23 to Openflow) and 332 (applying same analysis to claim 1). Li discloses that “instead of a centralized controller, the control functionality may be distributed to the switches or base stations, where each switch can run an agent that performs simple control functions, thereby partitioning control functionality between the controller and the agent (i.e. distributed applications) [page 4, column 2]).” <i>Id.</i>, 328 (mapping corresponding “distributed applications” feature in independent claim 22 to Li) and 332 (applying same analysis to claim 1).
the programmable network device and programmable cloud device forming unified capabilities enabling a plurality of upper layer application programming interfaces (APIs) that allow for programming of the programmable network device and programmable cloud device independent of the programmable	Li discloses “ <i>remote control of virtualized base station resources using open API between the controller and base station [page 2, column 2]; OpenFlow switches can be used as base stations, that can be controlled using the OpenFlow protocol (i.e. upper layer API) [page 4 last paragraph]; SDN radio hardware exposes a well-defined API which can be controlled by the control plane [page 5, column 1, last paragraph]; vendor-specific configuration interfaces can be overcome using SDN (i.e. virtualization, implementing network functions in software (SDN) and using “open” APIs (OpenFlow), all enable hardware-independent</i>

network device and the programmable cloud device;	<i>control</i>) [page 2, first paragraph]. ... [T]he OpenFlow white paper discloses an example where “AmyOSPF” is a new routing protocol, AmyOSPF or any other custom routing protocol can be run in the controller (i.e. controller device programmed with AmyOSPF), and the controller adds flow-entries in the OpenFlow switches to perform the data-plane packet processing [page 4, column 1]. Therefore, it would have been obvious to a person of ordinary skill in the art, before the effective filing date of the claimed invention, to include “programming of the cloud device”, as taught by OpenFlow, in order to include the details of an exemplary OpenFlow implementation in the generic SDN discussion of Li.” <i>Id.</i>, 328-329 (mapping corresponding features in independent claim 22 to Li and OpenFlow) and 332 (applying same analysis to claim 1).
an infrastructure application market place in communication with an application management portal coupled to the programmable network device and programmable cloud device, said infrastructure market place capable of providing tested and certified distributed applications to the application management portal for installation in the first network applications and second network applications and allow the application management portal to manage life	“[A]nalogous art by Shatzkamer discloses app store portal for deploying virtualized network functions [Title]; support Network Function Virtualization (NFV) [0005] by providing a digital distribution platform or an “app store” (i.e. marketplace) for deployment of virtual network functions (VNFs) [0019], vendors or third-party developers communicate with the app store to submit VNFs according to prescribed certification requirements [0020]; provide a standardized web based application portal (i.e. application management portal) that can provide point-and-click deployment of the third-party VNFs [0021]. Provide a coordinated execution of virtualized network services by an by an orchestrator [0023]; the orchestrator includes a service provisioning module and a life-cycle management controller [0028]; for example, for a new evolved packet service (EPS), the orchestrator can allocate a virtualized eNodeB, a virtualized

cycles of the distributed applications;	MME, a virtualized SGW, a virtualized PGW, etc. [0031]. Therefore, it would have been obvious to a person of ordinary skill in the art, before the effective filing date of the claimed invention, to modify the software defined cellular network of Li that utilizes the OpenFlow protocol (Li-OpenFlow), to include an application marketplace and an application management portal as taught by Shatzkamer, because a standardized web-based app store portal enables a “point-and-click” deployment of third-party virtual network functions, thereby automating the complex tasks of commissioning, capacity planning and management of virtualized network services [Shatzkamer 0001, 0004].” <i>Id.</i>, 330 (mapping corresponding features in independent claim 22 to Li, OpenFlow and Shatzkamer) and 332 (applying same analysis to claim 1).
wherein the application management portal is further capable of live upgrading of the distributed applications with substantially no interruption to operation.	“[T]he underlying NFV white paper discloses that one of the benefits of NFV is to support in-service software upgrade, without interrupting service [NFV, page 9]. Therefore, it would have been obvious to a person of ordinary skill in the art, before the effective filing date of the claimed invention, to modify the combined disclosure of Li-OpenFlow-Shatzkamer to include “upgrade the distributed applications with substantially no interruption to operation” as recited in NFV white paper, in order to include a recitation of the benefits of the NFV paradigm that is used in Shatzkamer.” <i>Id.</i>, 331 (mapping corresponding features in independent claim 22 to Li, OpenFlow, Shatzkamer and NFV whitepaper) and 332 (applying same analysis to claim 1).

<u>Elements from Claim 17</u>	<u>Examiner's Mapping To Prior Art</u>
The system of claim 1, wherein the at least one of the plurality of first network applications and at least one of the plurality of second network applications are given an access level to determine which APIs can be accessed.	“Draluk discloses software platforms provide application programming interfaces (APIs) to allow interaction between developer level processes and platform level processes. It is desirable to provide different levels of access to these APIs, for example, some trusted applications may be permitted access to a full set of APIs while other third party applications may be permitted access to a limited set of APIs [0002]. Therefore, it would have been obvious to a person of ordinary skill in the art, before the effective filing date of the claimed invention, to modify Li-OpenFlow-Shatzkamer-NFV to include “wherein the at least one of the plurality of first network applications and at least one of the plurality of second network applications are given an access level to determine which APIs can be accessed” as taught by Draluk in order to prevent misuse of protected data by untrusted applications [Draluk, 0015].” <i>Id.</i> , 343.

<u>Elements from Claim 18</u>	<u>Examiner's Mapping To Prior Art</u>
The system of claim 1, wherein the programmable network device is adapted to verify the authenticity and integrity of updates to the first network applications and the programmable cloud device is adapted to verify the authenticity and integrity of updates to the second network applications.	“Homiller discloses at Block 406, the integrity of the software upgrade package may be verified, and at Block 406, the authenticity of the software upgrade package may be verified [0036, Fig. 4]. Therefore, it would have been obvious to a person of ordinary skill in the art, before the effective filing date of the claimed invention, to modify Li-OpenFlow-Shatzkamer-NFV to include verifying authenticity and integrity of deployed applications as taught by Homiller in order to check the upgrade package for errors, and to prevent the use of a software upgrade package that is intended for a different type of device, or prevent the use of software that has been developed, or modified, by a non-authorized individual or entity [Homiller, 0036].” <i>Id.</i> , 344.

2. First Examiner Interview and Response To Rejection

After the rejection, the Applicant conducted an interview with the Examiner to discuss the prior art applied in the rejection. *Id.*, 405-406. During the interview, the Applicant presented an amended version of claim 1 to the Examiner, but no agreement was reached. *Id.*, 409-410. After the interview, the Applicant filed a response which included an amended version of claim 1 which is set forth below in color-coded format. With one exception (discussed below) which turned out to be immaterial to the Examiner's decision to allow the case, each proposed amendment to claim 1 was substantially directed to previously rejected subject matter in that each such amendment either (i) rearranged within claim 1 similar limitations from the previously rejected version of the claim, or (ii) incorporated limitations from the previously rejected versions of dependent claims 17 and 18 into claim 1:

1. (Currently Amended) A system for processing data packets in a mobile data network that includes a base station coupled to a core network having a plurality of network elements, said plurality of network elements comprising:

a programmable network device adapted configured to host a plurality of first network applications and located at the wireless edge of the mobile data network;

a programmable cloud device adapted configured to host a plurality of second network applications and located at a backend location which allows for control of the first and second network applications across the mobile data network, wherein the plurality of first network applications in the programmable network device and the plurality of second network applications in the programmable cloud device are in secure communication with each other to form distributed applications;

the programmable network device and programmable cloud device forming unified capabilities enabling a plurality of ~~upper layer~~ application programming

interfaces (APIs) that allow for programming of the programmable network device and programmable cloud device independent of the programmable network device and the programmable cloud device, wherein the at least one of the plurality of first network applications and at least one of the plurality of second network applications are given an access level to determine which APIs can be accessed;

an application management portal coupled to the programmable network device and programmable cloud device, wherein the application management portal is coupled to an application repository which is capable of providing the distributed applications which have been tested and certified to the application management portal for installation in the programmable network device and programmable cloud device;

wherein the application management portal is further configured to manage life cycle of the distributed applications, and conduct live upgrades of the distributed applications with substantially no interruption to operation; and

wherein the programmable network device is adapted to verify the authenticity and integrity of upgrades to the first network applications and the programmable cloud device is adapted to verify the authenticity and integrity of upgrades to the second network applications

an infrastructure application market place in communication with an application management portal coupled to the programmable network device and programmable cloud device, said infrastructure market place capable of providing tested and certified distributed applications to the application management portal for installation in the first network applications and second network applications and allow the application management portal to manage life cycles of the distributed applications;

wherein the application management portal is further capable of live upgrading of the distributed applications with substantially no interruption to operation.

Id., 424-425. In amended claim 1 above, the underlined language highlighted in dark gray was incorporated from previously rejected dependent claim 17; the underlined language highlighted in light gray was incorporated from previously rejected dependent claim 18; and the underlined language highlighted in yellow, green, blue

and purple each respectively corresponds to similar language shown in strikethrough format later in the claim and highlighted in the same color, i.e., yellow, green,² blue and purple, respectively. The added limitations that were not substantially set forth in a previously rejected claim corresponded to the language highlighted in red, which required that the programmable network and cloud devices be located at the edge and backend of the network, respectively. As discussed below, the Examiner found that these added limitations (highlighted in red) were present in the previously applied references. In the remarks accompanying amended claim 1, the Applicant stated:

By this amendment, Applicant has added limitations to independent claim 1 in the following manner.

First, claim 17 has been cancelled and its limitations moved into claim 1. ...

Claim 17 was rejected under Claim Li-OpenFlow-Shatzkamer-NFV in view of Draluk.

Second, claim 18 has been cancelled and its limitations also moved into claim 1. ... Claim 18 was rejected over Li-OpenFlow-Shatzkamer-NFV in view of Homiller.

Third, claim 1 [h]as been amended to recite the following with the new language to the claim underlined: “a programmable network device ... located at the wireless edge of the mobile data network; a programmable cloud device ... located at a backend location ...”. ... By this Amendment, Applicant clarifies that the programmable network device is located at the wireless edge of the network with visibility into its portion of the network and the programmable cloud device is located

² In the revisions highlighted in green, the Applicant substituted “an application repository” for “an infrastructure application market place.”

a backend location in the network which allows it to control the distributed applications across the entire network. ... The Open Flow reference illustrates a centralized control architecture where a remote switch is controlled by a centralized controller. ... Open Flow does not allow for switches to directly communicate with each other. ...

Applicant respectfully submits that these amendments to claim 1 place the claim in condition for allowance. *Id.*, 436-437.

3. Second Examiner Interview

The Applicant then had a further interview with the Examiner during which the Examiner stated that the added limitations requiring that the programmable network and cloud devices be located at the edge and backend of the network, respectively, were not persuasive. In this regard, the Interview Summary stated:

We also discussed Applicant's Remarks filed on 8/31/201, and the Examiner indicated that the arguments are *not persuasive*, because Open Flow is not the primary reference, whereas the primary reference by Li clearly discloses SDN techniques for LTE (wireless) networks. Further, the argument about "allow for switches to directly communicate with each other" is moot because the feature is not claimed. The Applicant agreed with the Examiner's observations. *Id.*, 470 (emphasis added).

See also, id., 451-452 (rejecting arguments presented in prior response).

4. Examiner's Amendment

After the second interview, the Applicant and the Examiner exchanged "subsequent email communications" with respect to the language of the claims, and the Applicant authorized an Examiner's Amendment to claim 1 (set forth below in color-coded format) that, in the Examiner's view, put the claim in condition for allowance. *Id.*, 453.

1. (Currently Amended) A system for processing data packets in a mobile data wireless network, ~~that includes a base station coupled to a core network having a plurality of network elements~~ the system comprising:

a programmable network device ~~configured to host~~ hosting a plurality of first network applications and located at ~~the~~ a wireless edge of the mobile data wireless network;

a programmable cloud device ~~configured to host~~ hosting a plurality of second network applications and located in a core of the mobile data wireless network; ~~at a backend location which allows for control of the first and second network applications across the mobile data network~~;

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 to form a distributed application applications;

the programmable network device is powered by a sandboxing operating system which facilitates deployment of the plurality of first network applications independent of hardware vendor associated with the programmable network device, and facilitates hot upgrade of the plurality of first network applications with substantially no interruption to operation of the programmable network device;

~~forming unified capabilities enabling a plurality of application programming interfaces (API) that allow for programming of the programmable network device and programmable cloud device independent of the programmable network device and the programmable cloud device, wherein the at least one of the plurality of first network applications and at least one of the plurality of second network applications are given an access level to determine which APIs can be accessed;~~

wherein the programmable network device verifies authenticity and integrity of upgrades to the plurality of first network applications and the programmable cloud device verifies the authenticity and integrity of 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;

wherein the programmable network device and the programmable cloud device form unified capabilities using a plurality of application programming interfaces (APIs) wherein the APIs accessible to each of the plurality of the first network applications and the plurality of second network applications are determined based on a respective access level associated with each of the plurality of first and second network applications;

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

an application management portal coupled to the programmable network device, and the programmable cloud device and the application repository, wherein the application management portal presents a user interface to manage provisioning, usage and de-provisioning of the distributed applications on the programmable network device and the programmable cloud device ~~is coupled to an application repository which is capable of providing the distributed applications which have been tested and certified to the application management portal for installation in the programmable network device and programmable cloud device;~~

~~wherein the application management portal is further configured to manage a life cycle of the distributed applications, and conduct live upgrades of the distributed applications with substantially no interruption to operation;~~ and

~~wherein the programmable network device is adapted to verify the authenticity and integrity of upgrades to the first network applications and the programmable cloud device is adapted to verify the authenticity and integrity of upgrades to the second network applications.~~

Id., 453-455. In amended claim 1 above, the underlined language highlighted in yellow, green, blue, purple and dark grey each substantially corresponds to language shown in strikethrough format elsewhere in the claim and highlighted in the same color, i.e., yellow, green, blue, purple and dark grey, respectively. Thus, it is implausible that these changes were material to the Examiner's decision to allow the case. In fact, there were only two added limitations that were not substantially set forth in a previously rejected claim, namely, the language highlighted in red, and light grey above.

Focusing first on the new language in light grey above requiring that verification be “based on unique security keys associated with each of the plurality of first and second network applications,” the Examiner understood that this feature was in the prior art. More specifically, this feature (requiring verification based on keys) was disclosed in [0036] of Homiller -- the very paragraph of Homiller that the Examiner applied in his prior rejection. *Id.*, 344, citing Homiller at [0036] (“Homiller discloses at Block 406, the integrity of the software upgrade package may be verified ... [0036, Fig. 4]”) and Ex. 2006, [0036] (Homiller) (“At Block 404, the integrity of the software upgrade package may be verified, ... The determination of authenticity may be responsive to a comparison of information, such as an authorization key, with an authentication code. ... The key or code may indicate a software version and/or source of the software ... The authorization key ... may be separately stored within the initial version of software and the software upgrade package”). Given this disclosure in the paragraph of Homiller relied on by the Examiner, it is implausible that the Examiner based his decision to allow the application on the requirement that verification be based on keys.

By contrast, the new language in red above -- requiring that the programmable network device be “powered by a sandboxing operating system which facilitates deployment of the plurality of first network applications independent of hardware vendor associated with the programmable network device” -- was plainly not

disclosed in Shatzkamer³ (or any of the references applied by the Examiner), and provided a clear rationale for the Examiner's decision to allow the case. In fact, the Examiner emphasized the importance of the "sandboxing operating system" limitation in his reasons for allowance, which provided as follows:

The following is an examiner's statement of reasons for allowance:

The currently amended claims recite features that are not disclosed or suggested by the prior art of record. For instance, the prior art does not disclose forming distributed applications between a network application hosted by a network device at the edge of the wireless data network (e.g. base station or eNodeB) and a network application hosted by a cloud device in the core of the wireless data network (e.g. MME, SGW, PGW), wherein the network devices are powered by a sandboxing OS that facilitates deploying the applications in a hardware-independent manner and facilitates hot upgrades of the applications, wherein devices verify authenticity and integrity of the deployed applications based on the security keys assigned to the applications. Even though the Shatzkamer reference discloses an application repository that stores certified virtual network functions for deployment on physical machines via a web-based portal, Shatzkamer does not suggest storing and provisioning distributed applications as claimed. Therefore, the prior art of record does not disclose or suggest the subject matter presented in the independent claims as whole. *Id.*, 468 (emphasis added).

By way of summary, the prosecution history of the '624 Patent as a whole demonstrates that the Examiner's rationale for allowing of the '624 Patent was based on an Examiner's amendment requiring that the programmable network device

³ Shatzkamer's disclosure is discussed in Section VIII, below.

previously recited in the claims be narrowed to a programmable network device “powered by a sandboxing operating system.” The prosecution history as a whole demonstrates that, in the Examiner’s view, that every other limitation in claim 1 (as amended by the Examiner) was disclosed in the prior art applied in the prior rejection of the claim.

5. Prosecution History of ‘095 Patent

The ‘095 Patent is a continuation of U.S. Patent No. 10,686,871, which is a continuation of the ‘624 Patent. The claims of the ‘095 Patent carry forward from the ‘624 Patent the requirement of a programmable network device “powered by a sandboxing operating system.”⁴ As shown in the comparison chart below, the claims of the ‘095 Patent substantially track those of the ‘624 Patent with the exception that the limitation reciting “unified capabilities” that use “a plurality of application programming interfaces (APIs)” from the ‘624 Patent has been removed in the ‘095 Patent:

‘095 Patent - Claim 1	‘624 Patent - Claim 1
1. A system for processing data packets in a network, the system comprising:	1. A system for processing data packets in a mobile data wireless network, the system comprising:

⁴ As mentioned previously, since the ‘624 Patent and the ‘095 Patent derive from the same initial application, the prosecution history regarding a claim limitation in the ‘624 Patent applies with equal force to the same claim limitation in the ‘095 Patent. *Elkay Mfg. Co. v. Ebco Mfg. Co.*, 192 F.3d 973, 980 (Fed. Cir. 1999).

a programmable network device hosting a plurality of first network applications;	a programmable network device hosting a plurality of first network applications and located at a wireless edge of the mobile data wireless network;
a programmable cloud device hosting a plurality of second network applications;	a programmable cloud device hosting a plurality of second network applications and located in a core of the mobile data wireless network;
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;	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;
the programmable network device and programmable cloud device are each powered by <u>a sandboxing operating system</u> which facilitates deployment of the plurality of first and second network applications	the programmable network device is powered by <u>a sandboxing operating system</u> which facilitates deployment of the plurality of first network applications independent of hardware vendor associated with the programmable network device,
and facilitates upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device;	and facilitates hot upgrade of the plurality of first network applications with substantially no interruption to operation of the programmable network device;
wherein the programmable network device verifies authenticity of the upgrades to the plurality of first network applications	wherein the programmable network device verifies authenticity and integrity of upgrades to the plurality of first network applications
and the programmable cloud device verifies authenticity of the upgrades to	and the programmable cloud device verifies the authenticity and integrity of

the plurality of second network applications	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;	and wherein the verification is based on unique security keys associated with each of the plurality of first and second network applications;
	wherein the programmable network device and the programmable cloud device form unified capabilities using a plurality of application programming interfaces (APIs)
	wherein the APIs accessible to each of the plurality of the first network applications and the plurality of second network applications are determined based on a respective access level associated with each of the plurality of first and second network applications;
an application repository storing distributed applications which have been tested for installation in the programmable network device and programmable cloud device; and	an application repository storing distributed applications which have been tested and certified for installation in the programmable network device and programmable cloud device; and
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.	an application management portal coupled to the programmable network device, the programmable cloud device and the application repository, wherein the application management portal presents a user interface to manage provisioning, usage and de-provisioning of the distributed applications on the programmable network device and the programmable cloud device.

During prosecution of the ‘095 Patent, the Examiner discussed a double patenting rejection of the claims in view of, *inter alia*, the ‘624 Patent. Ex. 1002, 35.

The Applicant obviated the double patenting rejection by filing a terminal disclaimer and the Examiner subsequently allowed the application. *Id.*, 032.

As noted above, the prosecution history of the ‘624 Patent as a whole, demonstrates that the recitation of a programmable network device “powered by a sandboxing operating system” was critical to allowance of the claims. It follows based on the similarities between the claims of the ‘624 and ‘095 Patents (as underscored by the Examiner’s rejection of the ‘095 Patent for double patenting over the ‘624 Patent) that the recitation of the “sandboxing operating system” in the ‘095 Patent was also critical to the allowance of that patent.

IV. CLAIMS OF THE ‘095 PATENT

The ‘095 Patent includes three independent claims, namely claims 1, 15 and 17.⁵ For purposes of this Patent Owner Preliminary Response, claim 1 is representative and recites:

1. A system for processing data packets in a network, the system comprising:
 - a programmable network device hosting a plurality of first network applications;
 - a programmable cloud device hosting a plurality of second network applications;
 - 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

⁵ With respect to the independent claims, only independent claims 1 and 15 are challenged herein.

with each other to form a distributed application;
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 facilitates upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device;
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;
an application repository storing distributed applications which have been tested for installation in the programmable network device and programmable cloud device; and
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.

Three aspects of independent claim 1 are worthy of note.

First, claim 1 recites that the programmable network device and programmable cloud device are each powered by a sandboxing operating system. As explained above, the recited sandboxing operating system was critical to the allowance of the '095 Patent and corresponds, for example, to fxOS 302a disclosed in the specification. See, Section II(C) above. The sandboxing operating system disclosed and claimed in the '095 Patent provides operating system (OS) level

sandboxing and the Petition has failed to explain how this limitation is met by application-level isolation like that provided by Shatzkamer and considered during prosecution.

Second, claim 1 recites that the sandboxing operating system facilitates upgrades to the applications “with substantially no interruption to operation of the programmable network device and programmable cloud device.” To the extent the Petition advances any construction at all for this limitation, the Petition states that “if the term is governed by means-plus-function,” the following describes the structure of the recited functions: “[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” Petition, 12 (emphasis added). As explained below, the Petition nowhere identifies any disclosure in the relied on references that permits one instance of the application to remain running during the upgrade.

Third, claim 1 recites that “the application management portal manages usage of the distributed applications.”⁶ To the extent the Petition advances any construction

⁶ Independent claim 15 recites a similar limitation providing that “the application management portal manages provisioning, usage and de-provisioning of the distributed applications.”

at all for this limitation, the Petition states that “if the term is governed by means-plus-function,” the following describes the structure of the recited functions: “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.” Petition, 13 (emphasis added). As explained below, the Petition nowhere identifies the disclosure of a “manifest file” in the relied on references.

V. CLAIM CONSTRUCTION

The claim limitations at issue must be construed under *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005). Claim language is not read in isolation. *Phillips*, 415 F.3d at 1315. Rather, the claim language, when read in light of the specification and the prosecution history, elucidates the meaning of the claim language.

Under 37 C.F.R. § 42.104(b)(3), a petitioner must provide a clear articulation of “[h]ow the challenged claim is to be construed.” As explained more fully below, the Petitioner has failed to meet this burden by basing the Petition on claim constructions that Petitioner believes are incorrect, and falsely alleging “the Petition Grounds render the claims unpatentable under either party’s construction.” Petition, 11. While a Petitioner may not be foreclosed from taking different claim construction positions in the district court and the PTAB, a Petitioner choosing to do so is required to explain such inconsistencies and provide a justification for taking

different positions in different forums. Here, the Petition offers no justification for taking different and in some cases irreconcilable claim constructions in the district court and the PTAB. As a result, the Petition is deficient for failing to articulate how the challenged claims are to be construed under 37 C.F.R. § 42.104(b)(3). On this basis alone, institution should be denied.

In the district court, Petitioner has taken the position that key claim terms of the ‘095 Patent are indefinite or require specific technical structures, such as the presence of a “kernel” in a “sandboxing operating system.” Yet, in this IPR, the Petitioner has abandoned those positions in favor of interpretations intended to support invalidity arguments in this IPR—without acknowledging, let alone justifying, the inconsistency. The Board should not permit a petitioner to exploit the IPR process in this manner.

As detailed below, Petitioner’s indefiniteness allegations in the district court cannot be reconciled with Petitioner’s positions here. Petitioner’s failure to present a consistent and legally sound claim construction, and its reliance on claim interpretations it has elsewhere disparaged, provide ample basis for denial of institution.

A. In The District Court Petitioner Maintains That A “Sandboxing Operating System” Must Have A Kernel But In This Proceeding Petitioner Ignores Any Requirement That The “Sandboxing Operating System” Have A Kernel

Petitioner’s construction of a “sandboxing operating system” in the district court cannot be reconciled with Petitioner’s mapping of that term to the prior art (Vasell) in these proceedings. In the district court, Petitioner maintains that a “sandboxing operating system” is software that, *inter alia*, “contains a kernel.” Ex. 1072, 7-8 (emphasis added). However, in Petitioner’s mapping of Vasell to the claimed “sandboxing operating system,” the Petition makes no mention of a kernel. Pet., 53-55. In fact, the term “kernel” appears nowhere in the Petition, let alone in Vasell.

Petitioner asserts that “the Petition Grounds render the claims unpatentable under either party’s construction.” (Pet., 11) This assertion is demonstrably false, because the Petition does not allege, let alone demonstrate, that Vasell has a sandboxing operating system that includes a kernel. The purpose of an IPR is not to provide an accused infringer with a mechanism for simultaneously advancing a narrow claim construction in district court that suits Petitioner’s non-infringement arguments and a broad claim construction in the PTAB that suits Petitioner’s invalidity arguments. Such gamesmanship favors denial of the Petition.

B. Petitioner’s Indefiniteness Positions In The District Court Cannot Be Reconciled With Petitioner’s Interpretation Of The Claims Of The ‘095 Patent In This IPR

In the district court, Petitioner maintains that claim 1 of the ‘095 Patent is indefinite. For example, in the district court, Petitioner contends that the term

“powered by” in claim 1 is indefinite because the term “has no meaning to a POSITA from a technical perspective.” Ex. 1072, 13-15 (emphasis added). By contrast, in this IPR, Petitioner implies that such term has a “reasonable construction.” Petition, 10. These positions are diametrically opposed. A claim term with “no meaning to a POSITA” does not have a “reasonable construction.” Congress did not create the IPR process to provide an accused infringer with a mechanism for simultaneously advancing irreconcilable positions in different forums.

Petitioner’s indefiniteness positions in the district court are not limited to the term “powered by” but extend to at least four other terms in claim 1 of the ‘095 Patent. More specifically, in the district court, Petitioner contends that a POSITA would not have been able to understand the meaning of (i) “substantially no interruption,” (ii) “the application management portal [further] manages ... usage ... of the distributed applications...,” (iii) “unique security keys associated with each of the plurality of first and second network applications,” or (iv) “facilitates deployment.” Ex. 1072, 14-19. By contrast, in this IPR, Petitioner articulates no construction for these terms other than identifying corresponding structure “to the extent Patent Owner (“PO”) argues or the Board finds they are governed by means-plus-function.” Petition, 10-11, 13 (emphasis added). This approach fails to satisfy the requirement that the Petition (rather than the Board or Patent Owner) articulate in the first instance how the claim language is being constructed.

C. To The Extent The Petition Advances Any Construction For The Substantially No Interruption Claim Terms, The Petition’s Construction Requires Structure That Permits One Instance Of The Application To Remain Running During The Upgrade

To the extent the Petition articulates any construction for the Substantially No Interruption Claim Terms, the Petitioner has stated that Patent Owner concedes that the Substantially No Interruption Claim Terms are governed by a means plus function analysis under 35 USC §112(f) and, based on this so-called concession, the Petition identifies the following structure for the Substantially No Interruption Claim Terms: “[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.” Petition, 12 (emphasis added); Ex. 1072, 16. Thus, to the extent the Board finds that the Petition has advanced a construction for the Substantially No Interruption Claim Terms, Petitioner’s construction requires Petitioner to demonstrate structure in the relied on references that permits one instance of the application to remain running during the upgrade.

D. To The Extent The Petition Advances Any Construction For The Usage Management Claim Terms, The Petition’s Construction Requires Structure That Uses Information In A Secure Manifest File

To the extent the Petition articulates any construction for the Usage Management Claim Terms, the Petitioner has stated that Patent Owner concedes that

the Usage Management Claim Terms are governed by a means plus function analysis under 35 USC §112(f) and, based on this so-called concession, the Petition identifies the following structure for the Usage Management Claim Terms: “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.” Petition, 13 (emphasis added). Thus, to the extent the Board finds that the Petition has advanced a construction for the Usage Management Claim Terms, Petitioner’s construction requires Petitioner to demonstrate structure in the relied on references that uses information in a secure “manifest file.”

E. The Petition Fails To Articulate How The Challenged Claims Are To Be Construed

In summary, the Petitioner has failed to meet the requirement of 37 C.F.R. § 42.104(b)(3) to provide a clear articulation of how the challenged claim are to be construed. Instead, Petitioner bases the Petition on claim constructions that Petitioner believes are incorrect as evidenced by Petitioner’s filings in the district court. Further compounding these inconsistencies is Petitioner’s false allegation that “the Petition Grounds render the claims unpatentable under either party’s construction.” Petition, 11. Clearly, if a “sandboxing operating system” requires a kernel (as Petitioner has alleged in the district court), but neither the Petition nor the

reference relied on (Vasell) mention a kernel, the Petition Grounds fail to render the claims unpatentable under Petitioner's district court construction. Similarly, if the claims of the '095 Patent are indefinite (as Petitioner has alleged in the district court), the Petition Grounds fail to render the claims unpatentable under Petitioner's district court construction because the PTAB would be unable to construe the claims rendering the Grounds in the Petition moot. As a result, the Petition is deficient for failing to articulate how the challenged claims are to be construed and on this basis institution should be denied.

VI. APPLICABLE LEGAL STANDARDS

The Board may only grant a petition for IPR where "the information presented in the petition ... shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition." 35 U.S.C. § 314(a); 37 C.F.R. § 42.108(c). Accordingly, the Petitioner bears the burden of showing this statutory threshold has been met. *See* Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,756 (Aug. 14, 2012) ("The Board ... may institute a trial where the petitioner establishes that the standards for instituting the requested trial are met...").

VII. A PERSON HAVING ORDINARY SKILL IN THE ART

The Petition alleges that "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.” Petition, 14. For purposes of this Patent Owner Preliminary Response only, Patent Owner does not contest Petitioner’s proposed definition of a POSITA. However, Patent Owner reserves the right to contest Petitioner’s proposed definition of a POSITA should this challenge be instituted.

VIII. THE VIRTUALIZED APPLICATION RUNTIME ENVIRONMENT IN SHATZKAMER CONSIDERED DURING PROSECUTION

Before getting into the references cited in the Petition, Patent Owner first turns back to the Shatzkamer reference (Ex. 2001) considered during prosecution and correctly found wanting.

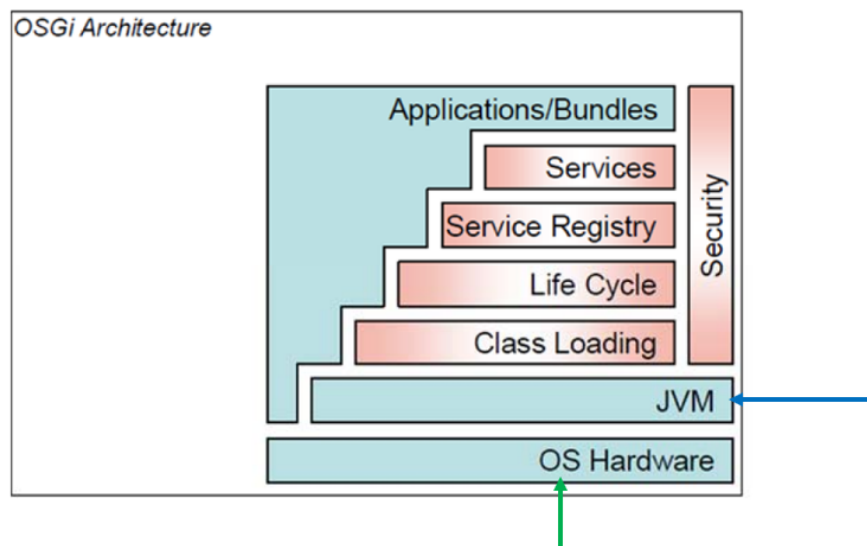
A. The Virtualized Runtime Environment Of Shatzkamer

The Shatzkamer reference applied by the Examiner disclosed a “virtualized application runtime environment” that could contain multiple instances of a network function (such as, e.g., Pandora or YouTube) running within the same environment. Ex. 2001, [0032] (disclosing operation of “one or more executable instances” of a network function within a “virtualized application runtime environment”) and [0071]-[0072] (describing applications such as Pandora and YouTube as examples

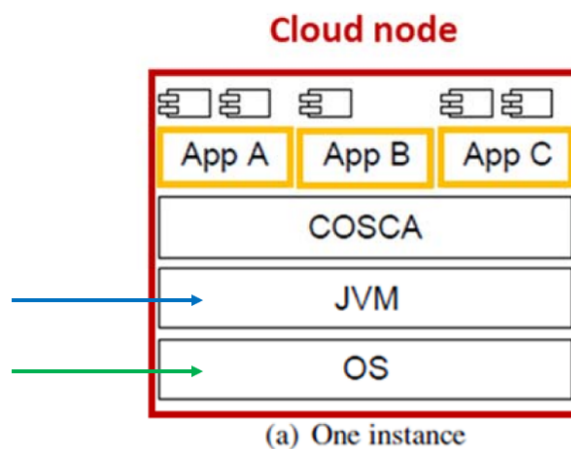
of network functions (or services)). A single runtime environment by itself, such as the one disclosed in Shatzkamer, does not provide isolation between different applications running in the runtime environment. Moreover, while a sandboxing operating system provides isolation at the operating system (OS) level, virtualized runtime environments by themselves, like that disclosed in Shatzkamer, achieve only application-level isolation. A virtualized runtime environment by itself provides only application-level isolation because, *inter alia*, the virtualized runtime environment sits at the level of an application running on top of the operating system and thus appears as such to the operating system. As discussed in the next section, Petitioner's expert has explained that a virtualized runtime environment like that of Shatzkamer is distinct from the operating system on which the virtualized runtime environment sits.

B. The Petition Explains That A Virtualized Runtime Environment Is Distinct From The Operating System On Which Virtualized Runtime Environment Sits

Petitioner's expert has explained that a virtualized runtime environment like that of Shatzkamer is distinct from the operating system on which the virtualized runtime environment sits. For example, in the following figure from the declaration of Petitioner's expert (the colored arrows have been added by Patent Owner), the Java Virtual Machine (JVM) (**blue** arrow) in the OSGi Architecture is distinct from the operating system (OS) (**green** arrow) upon which the JVM sits:



Ex. 1003, ¶194. Later in his declaration, as part of his “Technology Overview,” Petitioner’s expert again depicts a Java Virtual Machine (JVM) (blue arrow below) as being *distinct from the operating system* (OS) (green arrow) upon which the JVM sits:



Ex. 1003, ¶222 (colored arrows added by Patent Owner). Petitioner’s expert in fact describes the *JVM environment* as something “*running on the OS* a networked device.” Ex. 1003, ¶207 (emphasis added). Neither Petitioner, nor its expert, have

explained how a JVM runtime environment can be an operating system if the JVM “runs on” an operating system.

C. The Petition Conflates Isolation In General With Isolation At The Operating System Level As Recited In The Claims Of The ‘095 Patent

Petitioner’s expert opines that “software/process isolation, confinement, containment, and/or sandboxing occur in a variety of forms. (emphasis added). The common thread to this sandboxing is that there is always some layer logically below the isolated entities that enforces said isolation.” Ex. 1003, ¶116 (emphasis added). In the next sentence, Petitioner’s expert explains that an operating system (e.g., a sandboxing operating system) that “isolates processes and containers from each other” is one way to achieve isolation, while virtual machines provide a different way for achieving isolation. *Id.* Significantly, the claims of the ‘095 Patent do not generically claim every form of achieving isolation. Rather, the claims are limited to systems with a sandboxing operating system.

In summary, it is clear that the Shatzkamer reference that the Examiner applied disclosed only a “virtualized application runtime environment,” and, according to Petitioner’s expert, such a virtualized runtime environment is distinct from the operating system on which the virtualized runtime environment sits. The Petition has failed to explain how a “virtualized application runtime environment” (as disclosed, e.g., in Shatzkamer) that sits at the level of an application on top of an

operating system can itself correspond to a “sandboxing operating system” as recited in the claims of the ‘095 Patent.

IX. PRIOR ART RELIED ON IN PETITION

Ground 1 (the sole ground of the Petition) relies on Vasell, Alves, Hall and Rellermeyer, discussed below. Petition, 23.

A. Vasell (Ex. 1004)

Vasell is the primary reference relied on in the Petition. Vasell discloses an application and communication platform for connectivity based on services. Ex. 1004, Title.

1. The Virtual Runtime Environment In Vasell Is Cumulative Of That Disclosed In Shatzkamer

The Petition has failed to explain how Vasell’s local JVM environment is any different from the virtualized runtime environment disclosed in Shatzkamer and found wanting during prosecution. The Petition has failed to explain how Vasell discloses anything beyond application-level isolation “within a single local operating environment such as a Java Virtual Machine (JVM), or the like.” Ex. 1004, 18:27-29 (emphasis added). The Petition repeatedly describes Vasell’s JVM as being “hosted by” or “running on” an operating system, and thus distinct from the operating system itself. See, e.g., Pet., 43 (alleging that Vasell discloses “ ‘a Java runtime in the form of a Java virtual machine (JVM)’ and an OS hosting the JVM”); and 53 (asserting that Vasell discloses a “service application component respectively

hosted on a JVM running on an OS"). Neither Petitioner, nor its expert, have explained how a JVM runtime environment can be an operating system if the JVM "runs on" an operating system. Thus, with respect to the "sandboxing operating system" that was critical to the patentability, the Petition has failed to explain how the Vasell reference relied on in the Petition is anything other than cumulative of the Shatzkamer reference applied by the Examiner and correctly found wanting.

2. Vasell Discloses A Process For Updating A Configuration During Which Operation Of The Service Platform Server Stops And Starts

Vasell discloses the updating of a service application configuration. Ex. 1004, 22:19-30. While Vasell characterizes its update process as one that can "be switched on the fly" and which permits service to "continue undisturbed while configuration updates take place," Vasell's configuration update process in fact stops the existing configuration of the service application and then starts the new configuration. In this regard, Vasell discloses the following:

The information stored in the cell table 94 allows the 20 service platform server 22 to be switched on the fly to a new configuration of a service application 70---72. For instance, the cell manager 93 may be notified of a new configuration of one of the service applications 70-72 by a service provider 34 of FIG. 2. The cell manager 93 will then update to the new configuration by stopping the existing configuration of the service application 70-72, if it is running, and starting the new configuration. In this way, service may continue undisturbed while configuration updates take place. Ex. 1004, 22:19-28 (emphasis added).

B. Alves (Ex. 1008)

Alves is directed to the Open Service Gateway Initiative (OSGi), and discloses how “OSGi technology can be used to write better software, and in particular, how OSGi can be used to write better platforms for the development of better software.” Ex. 1008, xviii.

C. Hall (Ex. 1009)

Hall is also directed to OSGi, and discusses the most common concepts, features, and mechanisms needed to work with OSGi technology.” Ex. 1009, xx.

D. Rellermeyer (Ex. 1011)

Rellermeyer is directed to “replication of services on distributed cloud platforms.” Ex. 1011, 1.

X. GROUND 1 FAILS TO RENDER ANY OF THE CHALLENGED CLAIMS UNPATENTABLE BECAUSE THE PETITION HAS FAILED TO DEMONSTRATE THAT THE RELIED ON REFERENCES DISCLOSE A SANDBOXING OPERATING SYSTEM

A. The Petition Has Failed To Demonstrate That Vasell Discloses A Sandboxing Operating System

Claim 1 recites that the programmable network device and programmable cloud device are each “*powered by a sandboxing operating system.*” This limitation was critical to the allowance of the ‘095 Patent and provides sandboxing at *the operating system (OS) level*. The Petition relies solely on Vasell for this limitation.

The Petition fails to explain how Vasell discloses a sandboxing operating

system as claimed. More specifically, the Petition has failed to explain how Vasell discloses anything other than application-level isolation implemented with a JVM, or how Vasell discloses sandboxing at the operating system (OS) level, as claimed. For example, the Petition relies on Vasell's disclosure of multiple "gates" which work "within a single local operating environment such as a Java Virtual Machine (JVM), or the like." Ex. 1004, 18:27-29 (emphasis added). Absent from the Petition is any explanation how these disclosures from Vasell disclose anything other than application-level isolation implemented using a JVM (as was the case with Shatzkamer.)

The Petition repeatedly describes Vasell's JVM as being "hosted by" or "running on" an operating system, and thus distinct from the operating system itself. See, e.g., Pet., 43 (alleging that Vasell discloses " 'a Java runtime in the form of a Java virtual machine (JVM)' and an OS hosting the JVM"); and 53 (asserting that Vasell discloses a "service application component respectively hosted on a JVM running on an OS"). Thus, according to the Petition, Vasell's virtualized runtime environment must be distinct from the operating system on which the virtualized runtime environment sits. Given the Petition's discussion that Vasell's virtualized runtime environment is distinct from the operating system, the Petition fails to explain how Vasell's virtualized environment is itself an operating system that implements sandboxing as required by the claims of the '095 Patent.

For the above reasons, it is clear that the Petition has failed to explain how Vasell discloses a device “powered by a sandboxing operating system” as recited in Claim 1 of the ‘095 Patent, and thus failed to demonstrate a reasonable likelihood of success with respect to claim 1.

Like claim 1, independent claim 15 of the ‘095 Patent requires devices “powered by a sandboxing operating system.” Again, the Petition relies on Vasell alone for this limitation, and the Petition is therefore deficient with respect to claim 15 for the same reason as claim 1.

Finally, the Petition has likewise failed to demonstrate a reasonable likelihood of success with respect to any of the challenged dependent claims because the Petition’s showing with respect to the “sandboxing operating system” recited in the base claim is deficient.

B. Petitioner’s Conclusory Assertion That The Art Presented In The Petition Describes The Claim Limitations Added To Gain Allowance Is False

The Petition sets forth the conclusory allegation that “the prior art presented describes the claim limitations that were added to gain allowance.” Pet., 82 (emphasis added). In an attempt to support this allegation, the Petition obfuscates by providing a cross-citation to “§V.A-§VII.A.6” of the Petition -- which corresponds to pages 6-79 of the Petition. Nowhere in those pages (or elsewhere) does the Petition identify any “claim limitations that were added to gain allowance.” In fact,

the discussion of the prosecution history in the Petition is limited to a single sentence which merely states: “The Challenged Claims received only a double patenting rejection, which was overcome by a terminal disclaimer.” Pet., 10. Simply put, the Petition contains no analysis whatsoever to support its incorrect position that the prior art presented in the Petition “describes the claim limitations that were added to gain allowance.”

XI. GROUND 1 FAILS TO RENDER ANY OF THE CHALLENGED CLAIMS UNPATENTABLE BECAUSE THE PETITION HAS FAILED TO EXPLAIN HOW VASELL MEETS THE SUBSTANTIALLY NO INTERRUPTION CLAIM TERMS

Claim 1 recites that *the sandboxing operating system facilitates* upgrades with “substantially no interruption.” To the extent the Petition articulates any construction for the Substantially No Interruption Claim Terms, the Petitioner has stated that Patent Owner concedes that these terms are governed by a means plus function analysis under 35 USC §112(f) and identified the following structure for the Substantially No Interruption Claim Terms: “[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” Petition, 12 (emphasis added); Ex. 1072, 16.

Thus, to the extent the Board finds that the Petition has advanced a construction for the Substantially No Interruption Claim Terms, under that construction Petitioner was required to demonstrate that Vasell includes structure that permits one instance of the application to remain running during the upgrade in the relied on references. The Petition has failed to make this showing. The Petition fails to even allege that Vasell discloses structure where one instance of the application remains running during the upgrade. As explained above, Vasell's configuration update process stops the existing configuration of the service application and then starts the new configuration. Ex. 1004, 22:19-28. Missing from the Petition is any explanation how Vasell's disclosure of stopping of an application followed by starting of an application includes an instance of an application remaining running during the upgrade. This deficiency by itself warrants denial of the Petition.

Moreover, even if the Petition had shown that, in Vasell's upgrade process, one instance of the application remains running during the upgrade (to be clear, this is not the case), the Petition also fails to demonstrate that such functionality is facilitated by the operating system as required by claim 1 of the '095 Patent. The passage in claim 1 reciting this requirement is set forth below:

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 facilitates upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device...

The Petition fails to explain how Vasell's starting/stopping is performed or facilitated by an operating system in Vasell. Vasell states that the updating function alleged to meet the subject limitation is performed by a "cell manager 93." Ex. 1004, 22:19-28. However, the Petition never alleges that Vasell's "cell manager 93" is part of an operating system, or otherwise explain how Vasell (which makes no mention of an "operating system") discloses an operating system that facilitates the upgrading functionality recited in claim 1.

For at least the these reasons, the Petition fails to demonstrate that Vasell discloses an operating system that facilitates upgrades to the applications "with substantially no interruption to operation" of the subject devices, as recited in claim 1 of the '095 Patent. Independent claim 15 of the '095 Patent recites a substantively identical limitation and the Petition is therefore deficient with respect to claim 15 for the same reason as claim 1. Finally, the Petition has likewise failed to demonstrate a reasonable likelihood of success with respect to any of the challenged dependent claims because the Petition's showing with respect to the base claim is deficient with respect to the Substantially No Interruption Claim Terms.

XII. GROUND 1 FAILS TO RENDER ANY OF THE CHALLENGED CLAIMS UNPATENTABLE BECAUSE THE PETITION HAS FAILED TO EXPLAIN HOW VASELL MEETS THE USAGE MANAGEMENT CLAIM TERMS

Claim 1 recites that “the application management portal manages usage of the distributed applications.” Claim 15 recites a similar limitation. See, note 1, *supra*. To the extent the Petition articulates any construction for the Usage Management Claim Terms, the Petitioner has stated that Patent Owner concedes that these terms are governed by a means plus function analysis under 35 USC §112(f) and identified the following structure for the Usage Management Claim Terms: “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.” Petition, 13 (emphasis added).

Thus, to the extent the Board finds that the Petition has advanced a construction for the Usage Management Claim Terms, under that construction Petitioner was required to demonstrate that Vasell includes a structure that uses information in a secure manifest file. The Petition has failed to make this showing. In fact, other than in connection with its discussion of claim construction, the Petition makes no mention of a “manifest file.” Again, this deficiency by itself warrants denial of the Petition.

For at least these reasons, the Petition fails to demonstrate that Vasell

discloses the Usage Management Claim Terms in claims 1 and 15. The Petition has likewise failed to demonstrate a reasonable likelihood of success with respect to any of the challenged dependent claims because the Petition's showing with respect to the base claim is deficient with respect to the Usage Management Claim Terms.

XIII. CONCLUSION

For the reasons set forth above, institution should be denied.

Respectfully submitted,

Dated: July 15, 2025

By: /Daniel H. Golub/
Daniel H. Golub
Registration No. 33,701

Counsel for Patent Owner
Edge Networking Systems, LLC

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this **PATENT OWNER'S PRELIMINARY RESPONSE** complies with the type-volume limits set forth in the Memorandum and 37 C.F.R. § 42.24 because it contains 12,414 words, excluding the parts of the Brief that are exempted by 37 C.F.R. § 42.24(a), according to the word processing system used to prepare this Brief.

Respectfully submitted,

Dated: July 15, 2025

By: /Daniel H. Golub/
Daniel H. Golub
Registration No. 33,701

Counsel for Patent Owner
Edge Networking Systems, LLC

CERTIFICATE OF SERVICE

The undersigned certifies that pursuant to 37 C.F.R. § 42.6(e), a copy of the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** was served via email (as consented to by counsel) on July 15, 2025 to lead and backup counsel of record for Petitioner as follows:

Christopher J. Tyson
DUANE MORRIS LLP
901 New York Avenue N.W., Suite 700 East, Washington, D.C. 20001
CJTyson@duanemorris.com

John M. Baird
DUANE MORRIS LLP
901 New York Avenue N.W., Suite 700 East, Washington, D.C. 20001
JMBaird@duanemorris.com

Glenn D. Richeson
DUANE MORRIS LLP
1075 Peachtree Street NE, Suite 1700, Atlanta, GA 30309
GDRicheson@duanemorris.com

Patrick D. McPherson
DUANE MORRIS LLP
901 New York Avenue N.W., Suite 700 East, Washington, D.C. 20001
PDMcPherson@duanemorris.com

Respectfully submitted,

Dated: July 15, 2025

By: /Daniel H. Golub/
Daniel H. Golub
Registration No. 33,701

Counsel for Patent Owner
Edge Networking Systems, LLC