

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

SANDPIPER CDN, LLC,
Patent Owner

Case IPR2025-01010
U.S. Patent No. 10,057,322

**PATENT OWNER PRELIMINARY RESPONSE
UNDER 37 C.F.R. § 42.107(a)**

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

I.	INTRODUCTION	1
II.	BACKGROUND	3
III.	THE '322 PATENT	6
	A. The '322 patent provides novel techniques for improving the performance of content delivery networks.	6
	B. Prosecution History.....	8
	C. Level of Ordinary Skill in the Art	11
	D. Claim Construction	11
IV.	OVERVIEW OF THE PRIMARILY RELIED ON REFERENCES	12
	A. Verma.....	12
	B. Raciborski.....	13
	C. Lewin.....	15
V.	PETITIONER FAILS TO SHOW THAT VERMA, ALONE OR IN COMBINATION WITH ANY OTHER REFERENCE, RENDERS OBVIOUS ANY CHALLENGED CLAIM.	16
	A. The Petition fails to establish that the cited art discloses or renders obvious the claimed “modified embedded resource” limitations (all Grounds, all independent claims).	17
	B. The Petition fails to establish that the cited art discloses or renders obvious the claimed “application programming interface (API)” (Grounds 1A/1B, 2, 4; Claims 11 and 15).	20
	C. The Petition fails to establish that Verma, alone or in combination with Lewin, renders obvious the “tag” limitations of independent claim 1 (Grounds 3-5).	26
VI.	CONCLUSION	31

PATENT OWNER'S UPDATED EXHIBIT LIST

Exhibit No.	Description
2001	“Cloud CDN: content delivery network,” <i>Google Cloud</i> , accessed at https://cloud.google.com/cdn (last accessed July 2, 2025)
2002	Interim Processes for PTAB Workload Management Memorandum (March 26, 2025)
2003	National Judicial Caseload Profile – United States District Courts (March 2025)
2004	<i>Intentionally Left Blank</i>
2005	<i>Sandpiper CDN, LLC v. Google LLC</i> , No. 2:24-cv-03951, Invalidity Contentions (April 22, 2025)
2006	<i>Sandpiper CDN, LLC v. Google LLC</i> , No. 2:24-cv-03951, ECF No. 82 (C.D. Cal. June 24, 2025)
2007	<i>Sandpiper</i> , No. 2:24-cv-03951, ECF No. 28 (C.D. Cal. Sept. 16, 2024)
2008	FAQs for Interim Processes for PTAB Workload Management, accessed at https://www.uspto.gov/patents/ptab/faqs/interim-processes-workload-management
2009	Interim Director Discretionary Process, accessed at https://www.uspto.gov/patents/ptab/interim-director-discretionary-process
2010	Declaration of Samuel H. Russ, Ph.D.
2011	<i>Curriculum Vitae</i> of Samuel H. Russ, Ph.D.
2012	Goodwin, M., “What is an API (application programming interface)?” accessed at https://www.ibm.com/think/topics/api (last accessed August 5, 2025)

Exhibit No.	Description
2013	“CDN Stands For Content Delivery Network: History and Evolution,” <i>BlazingCDN</i> , accessed at https://blog.blazingcdn.com/en-us/cdn-stands-for-content-delivery-network-history-and-evolution (last accessed August 1, 2025)
2014	“What is a CDN (Content Delivery Network)?” <i>Amazon Web Services</i> , accessed at https://aws.amazon.com/what-is/cdn/ (last accessed August 1, 2025)

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Abbott Labs. v. Andrx Pharms., Inc.</i> , 452 F.3d 1331 (Fed. Cir. 2006)	15
<i>In-Depth Geophysical, Inc. v. ConocoPhillips Co.</i> , IPR2019-00850, Paper 56 (PTAB Sept. 3, 2020)	30
<i>Intelligent Bio-Sys., Inc. v. Illumina Cambridge Ltd.</i> , 821 F.3d 1359 (Fed. Cir. 2016)	30
<i>Sisvel Int'l. S.A. v. Sierra Wireless, Inc.</i> , 81 F.4th 1231 (Fed. Cir. 2023)	15
<i>Virtek Vision Int'l ULC v. Assembly Guidance Sys., Inc.</i> , 97 F.4th 882 (Fed. Cir. 2024)	35
<i>Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.</i> , 200 F.3d 795 (Fed. Cir. 1999)	16, 17, 18

I. INTRODUCTION

U.S. Patent No. 10,057,322 (“the ’322 patent”) provides novel techniques for distributing content to end users via content delivery networks (CDNs). These techniques involve an alternative to using conventional domain name system (DNS) servers for edge selection by instead calling an API, or otherwise querying the CDN, when constructing a uniform resource locator (URL) for the resource to be delivered from the CDN, improving the performance of CDNs.

Petitioner presents five Grounds challenging the claims of the ’322 patent:

Ground	Art	Basis	Claims (Independent Bolded)
1A	Verma (EX1005)	§ 102	15
1B	Verma	§ 103	11, 12, 14, 15
2	Verma and Raciborski (EX1006)	§ 103	11, 12, 14, 15
3	Verma and Lewin (EX1007)	§ 103	1, 8, 10, 13
4	Verma, Raciborski, Lewin	§ 103	2, 4, 5, 7, 8, 13
5	Vermin, Raciborski, Lewin, Draï (EX1008)	§ 103	3, 6

Pet., 10. Petitioner fails to establish that any of the ’322 patent’s independent claims 1, 11, and 15 are rendered obvious under any presented Ground.

First, for all Grounds, the Petition fails to establish that Verma discloses or renders obvious “obtain[ing]” or “generating a modified **embedded** resource,” as required by every independent claim. The ’322 patent describes that the claimed “modified **embedded** resource” is a resource embedded within another resource, such as an HTML document. Although Verma generally describes rewriting URLs,

Verma never describes that the rewritten URL is “embedded” within an HTML document or any other resource.

Second, for Grounds 1A/1B, 2, and 4 (independent claims 11 and 15), the Petition fails to establish that Verma renders obvious “application programming interface (API).” Petitioner relies solely on Verma’s content broker for the recited “application programming interface (API)” across all Grounds, but Verma’s content broker is not and does not perform the function of the claimed “application programming interface (API),” as claims 11 and 15 require. This element is wholly missing from Verma.

Finally, for Grounds 3-5 (independent claim 1), the Petition fails to establish that Verma, alone or in combination with Lewin, renders obvious the claimed “tag” limitations. Petitioner acknowledges that Verma does not disclose the claimed “tag” limitations, instead relying on Lewin. Pet., 45-48. Lewin’s purported tags, however, are not used to “obtain ... a modified embedded resource” that “includes a direct link to a node within the content delivery network that can deliver content associated with the embedded resource,” as claim 1 requires.

For at least these reasons, the Board should deny institution of this proceeding.

II. BACKGROUND

The challenged patent arises from pioneering technology directed to delivering content across a network. Content delivery networks (CDNs) provide the critical services that enable content providers like Google to quickly deliver online content to millions of consumers simultaneously over the Internet. EX2010, ¶26. The technology underlying these CDNs was made possible through a series of inventions developed by a California start-up called Sandpiper Software Consulting, LLC, a predecessor of Patent Owner Sandpiper CDN, LLC (“Patent Owner” or “Sandpiper”).

In the early 1990s, the World Wide Web saw increasing adoption, becoming a household staple. This mass adoption led to data congestion issues from the ever-growing number of users seeking to simultaneously access Internet content. EX2010, ¶26. A typical computer server in the 1990s could only handle a limited number of simultaneous connections before becoming overloaded. Moreover, signals take time to traverse routers and move through content-congested internet cables. Consumers living far from the physical server(s) hosting content experienced sluggish load times and high latency due to problems such as overloaded servers, congested network segments, and geographic separation. EX2010, ¶27. In the mid-1990s, Andrew Swart and David Farber were among the first individuals to develop services that allowed content providers to distribute

their content over the Internet, while avoiding the common congestion and performance issues that plagued Internet transmission at that time. Sandpiper's solution was to deploy CDN servers around the world, replicate appropriate content from customers' origin servers to appropriate CDN servers, and transparently rendezvous end-users requesting that content to the "best" CDN server to deliver that content, all while providing their customers with control over their content and user experience. This service and its architecture was quickly imitated by many others in the industry. EX2010, ¶28.

Using solutions developed by Mr. Swart and Mr. Farber, end-users would connect to an optimal edge server—one that that was closer to them or that had available capacity. Distributing content across a network of servers alleviated data congestion issues, and allowing consumers to connect to edge servers located near them reduced latency. EX2010, ¶29. Mr. Swart and Mr. Farber developed and built systems and methods for propagating data from origin servers to edge servers (known as "caching") based on network demand and for seamlessly routing users to the optimal edge server with the correct content. Using caching to perform the replication process within the CDN framework flew in the face of conventional wisdom. EX2010, ¶29.

In 1996, Mr. Swart and Mr. Farber founded Sandpiper Networks Inc. to further develop and commercialize their novel concept for a content delivery

network. Sandpiper Networks was based in Thousand Oaks, California. There, the inventors designed and built a CDN ultimately referred to as “Footprint.” By at least May 24, 1996, the Sandpiper team had conceived techniques for delivering static and dynamic content, including streaming resources, such as audio and video, using Sandpiper’s CDN.

Sandpiper Networks labored not only to build and implement its CDN, but also to protect their groundbreaking innovation through the patent system. Recognizing that its invention could revolutionize content delivery worldwide, Sandpiper Networks (and its successor companies, including Patent Owner) filed numerous patent applications directed to its foundational CDN technology, including the patent challenged in this Petition.

By at least May 1998, Sandpiper Networks was caching content and delivering cached content to end users of content providers using its CDN. Sandpiper Networks’ first paying customer was the L.A. Times, which paid Sandpiper Networks to host the report of Independent Counsel Ken Starr on his investigation of President Bill Clinton (“the Starr Report”) beginning on September 11, 1998. Sandpiper Networks’ CDN was capable of caching, and was used to cache and deliver Internet resources including inter alia, pictures, text files, dynamic resources, and streaming multimedia resources.

In the early-to-mid 2000s, demand for CDNs exploded. EX2010, ¶30.

Sandpiper Networks continued to commercialize its technology and compete in this expanding market, doing business as various entities including Digital Island, Inc. and a company called “Level 3.” It also created Patent Owner, Sandpiper CDN, LLC. Other businesses also entered the market by misappropriating Sandpiper’s innovations and directly competing with Sandpiper for market share.

One such company is Google. Google built its own in-house CDN, which uses technology described and claimed by Sandpiper’s patents. Additionally, YouTube, which Google purchased in 2006, relies upon CDN technology to meet its enormous data streaming needs, as well as for Google’s video search results (powered by YouTube). EX2001. Google did not license this patented technology from Sandpiper, Level 3, and/or its predecessors. Accordingly, Sandpiper brought suit against Petitioner Google LLC (“Petitioner” or “Google”) in district court for patent infringement. Google filed IPR Petitions against all of Sandpiper’s asserted patents in response.

III. THE ’322 PATENT

A. The ’322 patent provides novel techniques for improving the performance of content delivery networks.

The ’322 patent is entitled “Network Address Resolution,” and it claims priority to U.S. Provisional Application No. 62/098,930 (“the ’930 application”), filed on December 31, 2014.

The '322 patent is directed to novel techniques for improving the performance of content delivery networks (CDNs), in some cases using an application programming interface (API) “to embed link resolution information within an HTML document or other content returned to a requesting client.” EX1001, 1:13-18. Aspects of the '322 patent “involve an alternative to using conventional domain name system (DNS) servers for edge selection by instead calling an API or otherwise querying the CDN when constructing a uniform resource locator (URL) for the resource to be delivered from the CDN.” *Id.*, 3:9-14. The solutions described in the '322 patent were unconventional at the time of the invention of the '322 patent and continue to offer numerous technical benefits. EX2010, ¶31.

For example, as users were able to access more content (e.g., television, movies, news, software updates) on networks, network providers sought alternatives to routing requests through DNS to select an edge server. EX1001, 1:39-50. The inventors of the '322 patent devised techniques for “bypassing DNS,” thereby reducing the number of components involved in servicing requests while improving response speeds when delivering content. EX1001, 2:45-49, 5:29-32, 6:60-7:1; EX2010, ¶32. And these techniques are recited in the claims of the '322 patent. For instance, Petitioner-chosen representative independent claim 15 recites:

A method comprising:

providing, at a content delivery network, an application programming interface (API);

at the API, receiving, from an origin system that is separate from the content delivery network, an embedded resource and at least one parameter associated with the embedded resource;

generating a modified embedded resource providing a direct link to a node within the content delivery network for obtaining content associated with the embedded resource;

delivering, by the content delivery network, the modified embedded resource to the origin system;

receiving, at the node, a request from a client device to obtain the content; and

providing the content to the client device.

EX1001, 12:5-20.

The solutions described and claimed in the '322 patent improve the functioning of a CDN, including, for example, offloading computational resource consumption away from a client-side DNS infrastructure, and instead having content retrieved from an optimal node in the network. EX2010, ¶32.

B. Prosecution History

The '322 patent was filed as U.S. Application No. 14/985,968 (“the '968 application”) on December 31, 2015. The '322 patent claims priority to U.S. Provisional Application No. 62/098,930, filed December 31, 2014. The '322 patent

received a thorough prosecution on the merits, including overcoming both §§ 101 and 102 rejections. EX1002, 15-18, 60-74, 95-108.

After receiving a non-final rejection, *see* EX1002, 95-108, Patent Owner amended the independent claims to clarify the architecture of the claimed systems and methods. EX1002, 56-59; EX2010, ¶¶34-35. For example, Patent Owner amended independent claim 12 (which issued as independent claim 11) as follows:

12. (Currently Amended) A content delivery ~~device~~ network comprising:
at least one processor in communication with a non-transitory computer readable medium including computer readable instructions for:
~~generating~~ providing an application programming interface (API);
at the API, receiving, from an origin system that is separate from the content delivery network, an embedded resource and at least one parameter associated with the embedded resource;
generating a modified embedded resource providing ~~information as to a location within a~~ a direct link to a node within the content delivery network for obtaining content associated with the embedded resource;
delivering, by the content delivery network, the modified embedded resource to the origin system;
receiving, at the node, a request from a client device to obtain the content; and
providing the content to the client device.

EX1002, 57-58.

Patent Owner also added independent claim 22 (which issued as independent claim 15) with similar limitations:

22. (New) A method comprising:

providing, at a content delivery network, an application programming interface (API);

at the API, receiving, from an origin system that is separate from the content delivery network, an embedded resource and at least one parameter associated with the embedded resource;

generating a modified embedded resource providing a direct link to a node within the content delivery network for obtaining content associated with the embedded resource;

delivering, by the content delivery network, the modified embedded resource to the origin system;

receiving, at the node, a request from a client device to obtain the content; and

providing the content to the client device.

EX1002, 59.

Patent Owner argued that the claimed architecture differed from the prior art (U.S. Publication No. 2014/0289319 to Richardson) because “Richardson require[d] the client to either (a) communicate with the DNS server of the CDN to determine the node that will be used to deliver the content; or (b) execute translation code to ask the CDN for translation information, which translation information may include the address of the node that will be used to deliver the content.” EX1002, 72-73. Richardson thus did not disclose “generating modified embedded resource providing a direct link to a node within the content delivery network for obtaining content associated with the embedded resource.” *Id.*;

EX2010, ¶36. Upon consideration of this and other arguments, the Examiner allowed the claims. EX1002, 15-18.

As discussed below, the prior art presented in the Petition also fails to disclose or render obvious the limitations of the independent claims found allowable by the Examiner after being amended by Patent Owner during prosecution. These failures warrant denial of the Petition.

C. Level of Ordinary Skill in the Art

A POSITA is presumed to be aware of all pertinent art. A POSITA possesses the “understandings and knowledge reflected in the prior art,” and thinks along conventional wisdom. *See Abbott Labs. v. Andrx Pharms., Inc.*, 452 F.3d 1331, 1336 (Fed. Cir. 2006). At this stage, Patent Owner submits that the Board does not need to resolve the appropriate level of skill for a POSITA for purposes of deciding institution because the Petition fails under any reasonable articulation of a POSITA.

D. Claim Construction

Absent lexicography or disavowal, claim terms in an IPR are given their plain and ordinary meaning in light of the specification and prosecution history. *Sisvel Int’l. S.A. v. Sierra Wireless, Inc.*, 81 F.4th 1231, 1236 (Fed. Cir. 2023). At this preliminary stage, no term requires express construction beyond its plain and ordinary meaning, as would have been understood by a POSITA at the time of the

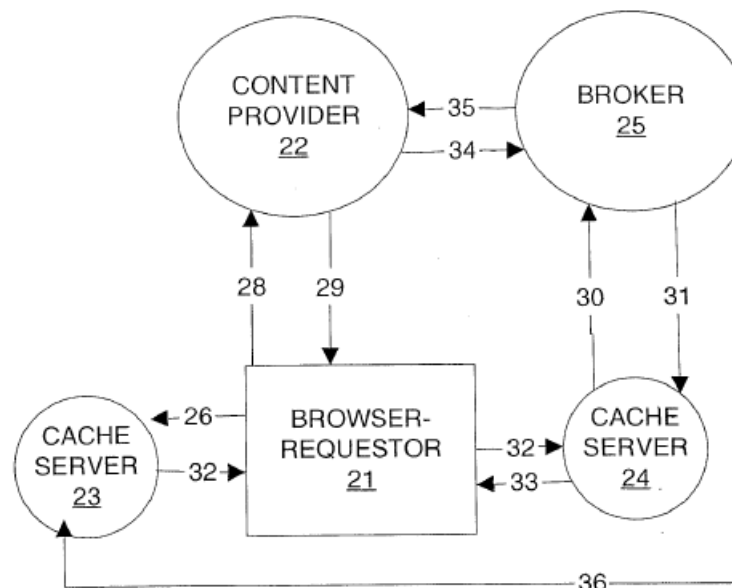
invention in light of the specification and the patent's prosecution history. *See Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

IV. OVERVIEW OF THE PRIMARILY RELIED ON REFERENCES

A. Verma

Verma discloses a “computer method and related apparatus for delivering content files to a client computer.” EX1005, Abstract, ¶¶10-11. The method includes modifying a resource link—based on client location and content availability—to direct requests to a cache server node in a CDN. EX2010, ¶39.

Figure 1 shows an exemplary “content delivery network.”



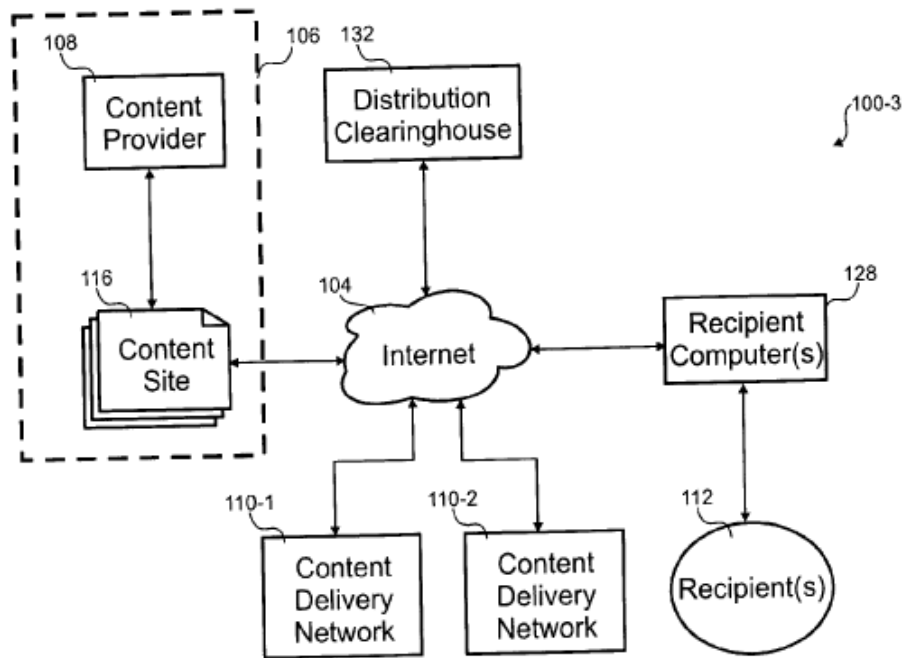
EX1005, FIG. 1.

In Verma, the content delivery network includes a client that runs a “browser 21,” a “content provider 22,” a “content delivery network broker 25,” and “a set of cache servers” “23”/“24.” EX1005, FIG. 1, ¶¶22, 25; EX2010, ¶¶40-41. The client

“browser 21 operates to request a content file ... by sending a URL 28 to a content provider 22.” EX1005, ¶25, FIG. 1. “The request from the client ... can comprise delay tolerance information” *Id.*, ¶18. This request is then “forward[ed]” from content provider 22 to the content delivery network broker 25 that “manages at least one, and preferably, a set of cache servers.” *Id.*, claim 9, ¶25. Also, content provider 22 can “dynamically request the broker 25 for a content file delivery to a location close to the client.” *Id.*, ¶26. The broker sends availability inquiries to the cache servers to “find out whether they currently store the requested” file. *Id.*, ¶25. The broker 25 can select the appropriate cache server to serve the content and return a “re-written URL to the content server [that] received the [client] request.” *Id.*, ¶¶12, 26-27, 37. When the client receives the prepended (re-written) URL, “it requests the content file from the appropriate cache server when the content is available.” *Id.*, ¶37; EX2010, ¶41.

B. Raciborski

Raciborski discloses a “content delivery network (CDN) for delivering content objects from a content provider to a recipient.” EX1006, Abstract; EX2010, ¶42. Figure 1C shows an exemplary network.



EX1006, FIG. 1C.

Within this network, Raciborski discloses a “content provider” that “sends various information to the one or more CDNs” in the network to facilitate content delivery. EX1006, ¶¶26, 55-56. The information can include the “IP address of the recipient computer,” the “full object location (i.e., server, path and name),” “if delivery can be delayed,” and a “protocol used to deliver the content object,” among other things. *Id.* This information may be included in a “bid message,” like the one illustrated in Table 1. *Id.*, ¶¶55-56, Table 1.

TABLE I

<u>Real Time Bid Message Example</u>	
Field	Example
Bid Identifier	34sd89732k
Recipient IP Address	69.93.18.222
Domain of Content Object	LLNW.com
Path of Content Object	/press/2005releases/
File Name of Content Object	routemark.mp4
Format	MPEG4
Bitrate	200 Kbps
Latency Options	3
QoS Level(s)	3
Packet Loss Options	2
Buffer Duration	24 hours
Max. Volume	1000

EX1006, Table 1.

Such “bid message” parameters, as illustrated in Table 1, provide the CDN with information it can use to select an optimal delivery path. EX1006, Table 1; EX2010, ¶43.

C. Lewin

Lewin is generally directed to a “dynamic content assembly mechanism” for constructing an “HTML page.” EX1007, ¶¶8, 12, 28, 49. As Lewin explains, “[t]raditionally,” “directives [] are placed in HTML pages” “that are evaluated on a server before the page is served.” *Id.*, ¶28. These “directives” or “dynamic assembly instructions” may be “URLs for HTML chunks to be incorporated in[to] the final HTML page.” *Id.*, ¶¶49, 58 (“dynamic content assembly page directives”). These “URLs” are “specif[ied]” by “tags” that can be identified when a server

“parses the HTML page.” *Id.* Of relevance, Lewin’s “tags,” such as the “<esi:include> tag” relied on by Petitioner, are used to instruct the browser to retrieve content at the URL—not to retrieve a link to the content, as the ’322 patent claims require. *Id.*, ¶30; EX2010, ¶44.

V. PETITIONER FAILS TO SHOW THAT VERMA, ALONE OR IN COMBINATION WITH ANY OTHER REFERENCE, RENDERS OBVIOUS ANY CHALLENGED CLAIM.

The Petition fails to establish a reasonable likelihood that any challenged claim of the ’322 patent is rendered obvious by the art, especially independent claims 1, 11, and 15.

First, for all Grounds and independent claims, Petitioner fails to establish that Verma discloses or renders obvious “obtain[ing]” or “generating a modified **embedded** resource,” as required by every independent claim. Petitioner relies on Verma’s purported disclosure of rewriting URLs, but Petitioner fails to establish that the rewritten URLs are “embedded” in any document or other resource, as the claims require. Petitioner’s mapping essentially reads the word “embedded” out of the claimed “modified embedded resource.”

Second, for Grounds 1A/1B, 2, and 4 (independent claims 11 and 15), the Petition fails to establish that the art renders obvious the claimed “application programming interface (API).” Petitioner relies solely on Verma’s content broker to satisfy this element, but Verma’s content broker is not an “API” as claimed.

Finally, for Grounds 3-5 (independent claim 1), the Petition fails to establish that Verma, alone or in combination with Lewin, renders obvious the claimed “tag” limitations. Petitioner relies on Lewin for the recited “tags,” but fails to show that Lewin’s purported tags are used to “obtain ... a modified embedded resource” that “includes a direct link to a node within the content delivery network that can deliver content associated with the embedded resource,” as required by claim 1.

The Petition’s flaws, therefore, span across all three independent claims 1, 11, and 15. Accordingly, the Petition does not establish a reasonable likelihood of success with respect to any claim. Institution should be denied.

A. The Petition fails to establish that the cited art discloses or renders obvious the claimed “modified embedded resource” limitations (all Grounds, all independent claims).

Independent claims 11 and 15 each recite, “generating **a modified embedded resource** providing a direct link to a node within the content delivery network for obtaining content associated with the embedded resource.” EX1001, 11:1-4 (element [11d]), 12:12-15 (element [15c]) (emphasis added). Claim 1 similarly recites, “obtain[ing], by the origin system, **a modified embedded resource** from a content delivery network.” *Id.*, 10:9-12 (element [1b]) (emphasis added); EX2010, ¶67.

The Petition relies on Verma for the recited “modified embedded resource” across all Grounds. Pet., 22-23, 52-56. Specifically, Petitioner alleges, “[t]he

broker's 'generat[ed] uniform resource locator' is a 'prepended'/'rewritten URL' because it takes an 'original URL' ('http://contentURL') and rewrites it into an "Akamai-ed" URL' identifying the local_cache_server in the path ('http://local_cache_server/contentURL') that redirects to a local cache server in the content delivery network." Pet., 22-23 (citing EX1005, ¶¶12-13, 16, 26). Petitioner fails to show, however, that Verma's "prepended" or "rewritten URL" is a "modified **embedded** resource" as claimed—e.g., as a modified resource embedded within the client's HTML page. EX2010, ¶69. Indeed, Petitioner's mapping reads the word "embedded" out of the claimed "modified embedded resource" altogether.

As the '322 patent describes, "when a user enters, into the browser Web page, an address http://www.example.com," the origin server returns an "HTML document 20 with an embedded resource 24." EX1001, 4:5-26. "The HTML document is what the browser uses to display the Web page [with an address] http://www.example.com" and "[t]he embedded resource is a link to some form of content that the user can obtain by selecting the link when the Web page is displayed." *Id.*, 4:26-31; EX2010, ¶68. The '322 patent further explains that "the HTML document may include an embedded resource" with the tag "\$" which, in turn, "**causes the HTML document returned to the requesting client to include a modified embedded resource 24B** with an address or URL to the CDN whereby

the content may be retrieved.” EX1001, 4:32-48 (emphasis added). Thus, as the ’322 patent describes and claims, the “modified **embedded** resource” is a resource embedded within another resource, such as an HTML document. *See, e.g.*, EX1001, 10:6-8, 10:56-58 (Claims 1 and 10: claiming a “content resource including at least one embedded resource, ... wherein the content resource is a hypertext markup language (HTML) document and the embedded resource is a link”), 11:10-16 (Claims 12 and 13 (“replacing the embedded resource with the modified embedded resource in a resource ... wherein the resource is an HTML document and the modified embedded resource is a link”)); EX2010, ¶68.

In contrast to the ’322 patent, Verma does not disclose embedding its “prepending” or “rewritten URL” within an HTML document returned to the client. EX2010, ¶70. Rather, Verma discloses automatically “redirecting to the cache server” when the URL is received, without embedding the URL within any other resource. EX1005, ¶37; EX2010, ¶70. Verma describes that the client “browser 21 operates to request a content file, for example the movie entitled ‘Matrix Reloaded,’ by sending a URL 28 to a content provider 22.” EX1005, ¶25. Then, as Petitioner recognizes, the “prepending URL redirecting to [a] cache server ... is returned to the client,” and “[w]hen the client receives the **prepending URL, it requests the content file from the appropriate cache server.**” EX1005, ¶37; Pet., 27; EX2010, ¶70. Indeed, Petitioner does not even allege that the “prepending”

or “rewritten URL” is embedded within an HTML document or other resource.

Pet., 22-23; EX2010, ¶70.

Accordingly, although Verma describes altering a URL, Petitioner fails to establish that Verma discloses or renders obvious “obtain[ing]” or “generating a modified **embedded** resource,” as required by independent claims 1, 11, and 15.

EX2010, ¶71.

B. The Petition fails to establish that the cited art discloses or renders obvious the claimed “application programming interface (API)” (Grounds 1A/1B, 2, 4; Claims 11 and 15).

Independent claims 11 and 15 recite various features involving an “application programming interface (API),” specifically:

[11b], [15a]: providing, at a content delivery network, **an application programming interface (API)**;

[11c], [15b]: **at the API**, receiving, from an origin system that is separate from the content delivery network, an embedded resource and at least one parameter associated with the embedded resource; ...

EX1001, 10:63-67 (Claim 11), 12:6-11 (Claim 15).

Dependent claims 2 and 9 also recite various features involving an “API,” specifically:

[2a]: passing information based on the client IP address to **an application programming interface (API)** of the content delivery network;

[2b]: **in association with the API**, obtaining a parameter associated with the embedded resource; ...

[9b]: passing, by the origin system, the location identifier to **an application programming interface (API)** of the content delivery network to generate the modified embedded resource relating to the node.

EX1001, 10:23-27 (Claim 2), 10:52-55 (Claim 9).

The Petition relies solely on Verma for the recited “application programming interface (API)” across all Grounds. Pet., 11-12. Although Grounds 2 and 4 further rely on Raciborski and Lewin for certain limitations of claims 2, 9, 11, and 15, Petitioner does not rely on either reference for the recited “application programming interface (API).” Pet., 34, 37-39 (relying on Raciborski to allegedly render obvious “receiv[ing] the embedded resource and parameter associated with the embedded resource at Verma’s ‘broker’ to achieve Verma’s goal”—not for the recited API), 54 (relying on Lewin to allegedly render obvious “using a delivery parameter to obtain, by the origin system, a modified embedded resource from the CDN ‘based upon identification of the tag’”), 67-68, 77; EX2010, ¶48.

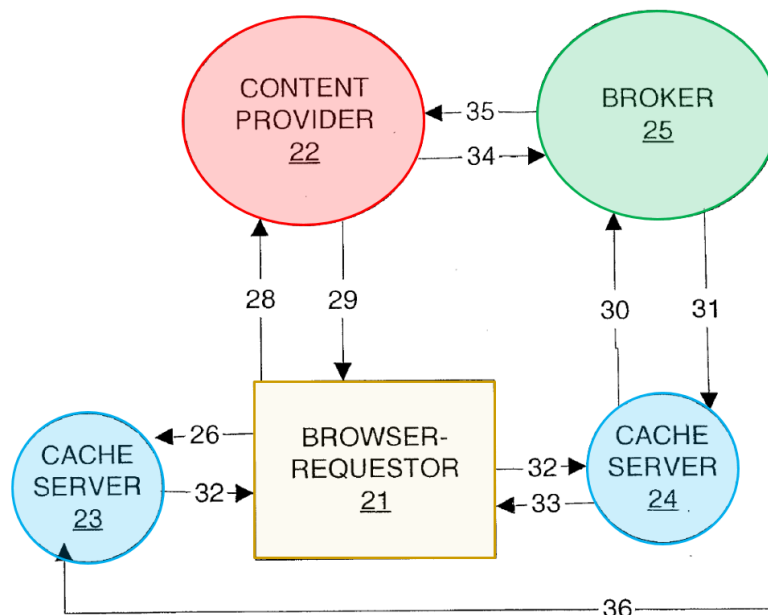
Specifically, Petitioner alleges, “a POSITA would have understood Verma’s ‘**broker**’ is an API because it performs the same logical functions as the API in the ’322 patent, which provides that ‘[w]hen the origin server receives the request,’ it ‘calls an API’ to ‘communicate with the compute engine’ in the CDN.” Pet., 12 (citing EX1001, 4:49-52, 7:16-25). Verma’s “broker,” however, is not an API. EX2010, ¶49.

As Patent Owner’s expert Dr. Samuel Russ, who has more than 25 years of experience in this field, explains, a POSITA would have understood an application programming interface (API) to be “a set of rules or protocols that enables software applications to communicate with each other to exchange data, features and functionality.” EX2010, ¶50 (citing EX2012, 2). For example, Dr. Russ explains that “in terms of a request and response between a client and server,” “[t]he API is the bridge establishing the connection between them.” EX2010, ¶50. In other words, as its name implies, an API acts as an interface that allows the client to communicate with an application executing on the server—it is *not* the application itself. EX2010, ¶50.

In the context of the ’322 patent, the “API” acts as a bridge between the origin server and the compute engine. EX1001, 6:24-30 (“[T]he API 26 provides the interface to a computing engine 30 that uses the information, passed to the engine through the API ...”), 4:32-34 (“[C]ompute engine 30 should be **contacted via a call to an API 26** (operation 220).” (emphasis added)); EX2010, ¶51. The ’322 patent further explains that “[w]hen the origin server receives the request (or at some time prior to receiving the request), the origin server **calls an API** in order to communicate with the compute engine 30 to format the address for the embedded resource.” EX1001, 4:49-52 (emphasis added); EX2010, ¶51. Then,

the “computing engine 30,” not the API, “determine[s] one or more nodes 32 in the CDN to serve the content” EX1001, 6:24-30; EX2010, ¶51.

Verma does not disclose or teach an API. EX2010, ¶52. In contrast to the claimed API, Verma describes “a content broker which receives information from the cache servers in a content delivery network as to traffic, availability of particular files and various locations, and then returns the re-written URL to the content server which received the request from the client.” EX1005, ¶12. For example, as depicted in Verma’s Figure 1, a “browser 21 operates to request a content file, for example the movie entitled ‘Matrix Reloaded,’ by sending a URL 28 to a content provider 22, Movies-Inc.Com in this example.” EX1005, ¶25; EX2010, ¶52.



EX1005, FIG. 1 (annotated).

As Verma explains, “Movies-Inc.Com content provider 22 sends a request 34 for information to a content delivery network broker 25, which manages at least one, and preferably, a set of cache servers, exemplified by cache server 23 and cache server 24.” EX1005, ¶25. The broker sends availability inquiries to the cache servers, respectively, “to find out whether they currently store the requested movie file.” *Id.* Verma explains that “[t]he broker 25 provides content current and/or future availability information to Movies-Inc.Com content provider 22 where the content file URLs are prepended (Akamai-ed) for redirecting to cache servers.” *Id.*, ¶26. But nowhere does Verma disclose that its “content broker” is, or makes use of, an API to communicate with either the content provider or cache servers. EX2010, ¶53.

As Petitioner acknowledges, the ’322 patent describes that an “API is a convenient and efficient way for disparate systems to uniformly and predictably **pass information to the computing engine and for the computing engine to return information.**” Pet., 12 (emphasis added). Verma’s purported “API,” however, performs entirely different logical functions. EX2010, ¶54. For example, Verma explains “[i]f the content file [to be delivered] is not immediately available ... Verma’s content server will negotiate with the content delivery network through the broker in order to determine the start time, end time, and the cache server where the content file will next be available during step.” EX1005, ¶38.

Determining the start time, end time, and the availability of the content file within each cache server—which are the logical functions performed by Verma’s broker—is different than an API that merely provides an interface for passing and returning information. EX2010, ¶54. Instead, Verma’s determination requires that its broker “schedule the distribution of content files to edge [or cache] servers.” EX1005, ¶22. EX2010, ¶54.

Moreover, the claims require that the “API” receive “at least one parameter associated with the embedded resource.” EX1001, 10:64-67 (Claim 11), 12:8-11 (Claim 15), 10:23-27, 10:36-37 (Claims 2 and 4 reciting similar limitations). But Verma is silent regarding the structure of the request sent from its content provider to the broker, including what information is passed. *See* EX1005, ¶¶25-26. Though Petitioner alleges that “delay tolerance information” is passed “by the content provider to, and received by, the broker,” Pet., 17, nowhere does Verma disclose this operation. Instead, Verma discloses that information is sent only from “the client” to the content provider. EX1005, ¶¶18, 25. And the broker “provides content current and/or future availability information” back to the content provider—regardless of the “delay tolerance information.” EX1005, ¶¶18, 26; EX2010, ¶54. Thus, Verma does not disclose nor require use of an “API” to receive “an embedded resource and at least one parameter associated with the embedded resource,” as claimed.

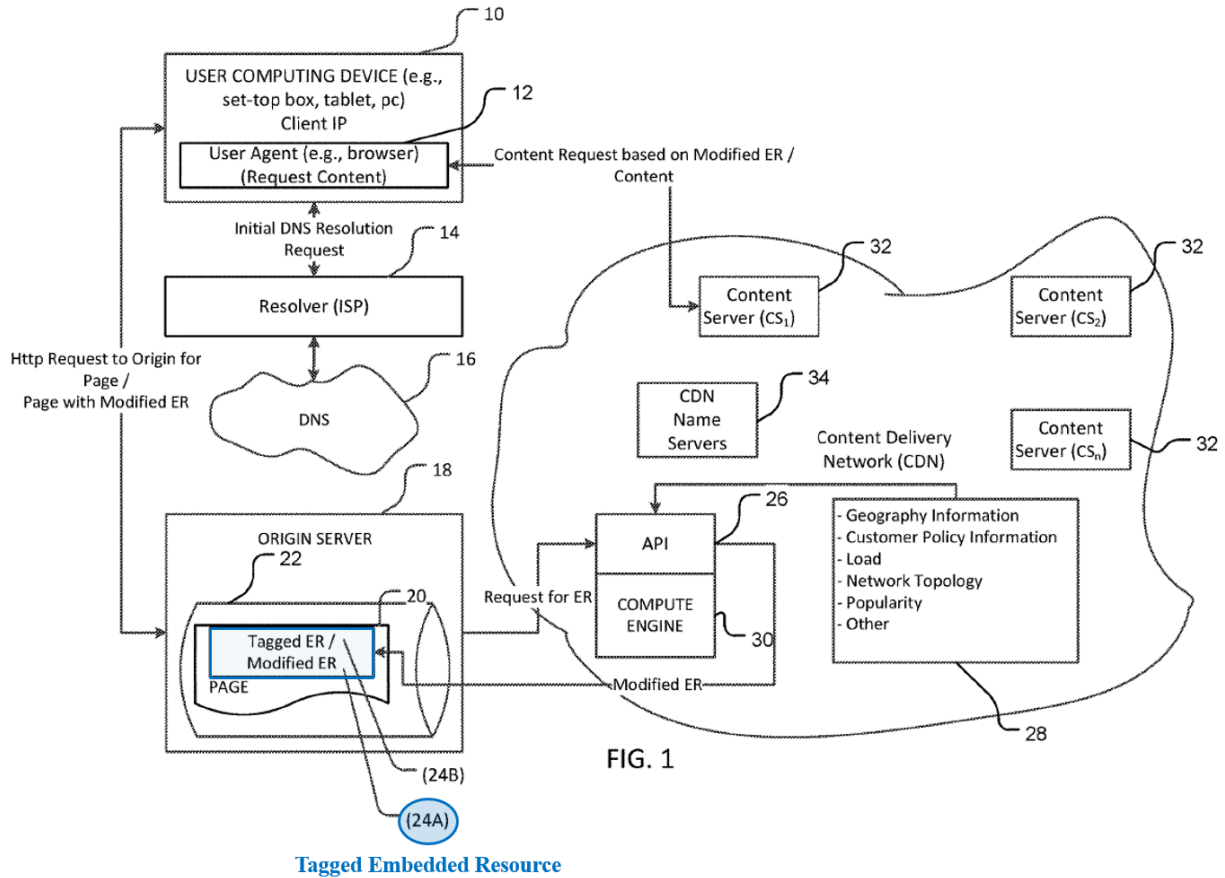
Accordingly, Verma does not teach that its broker is or provides the function of the claimed “application programming interface (API),” as required by claims 2, 9, 11, and 15. EX2010, ¶55. This element is wholly missing from Verma and is fatal to Petitioner’s analysis with respect to these claims. *In-Depth Geophysical, Inc. v. ConocoPhillips Co.*, IPR2019-00850, Paper 56 at 27 (PTAB Sept. 3, 2020) (Petitioner “must ‘specify where each element of the claim is found in the [relied upon] prior art patents.’”) (quoting 37 C.F.R. § 42.104(b)(4)); *Intelligent Bio-Sys., Inc. v. Illumina Cambridge Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016).

C. The Petition fails to establish that Verma, alone or in combination with Lewin, renders obvious the “tag” limitations of independent claim 1 (Grounds 3-5).

Independent claim 1 recites, “**receiving**, by an origin system, a request from a client device for a content resource, the content resource including **at least one embedded resource with a tag**,” and “**upon identification of the tag**, using at least one delivery parameter to **obtain, by the origin system, a modified embedded resource** from a content delivery network that is separate from the origin system.” EX1001, 10:6-12 (elements [1a] and [1b]) (emphasis added). Claim 1 further requires that “the **modified embedded resource includes a direct link to a node within the content delivery network** that can deliver content associated with the embedded resource.” *Id.*, 10:13-18 (element [1c]) (emphasis added); EX2010, ¶56.

Petitioner acknowledges that “Verma does not expressly disclose its origin system receiving ‘a request from a client device for a content resource, the content resource including at least one embedded resource with a tag,’” instead relying on Lewin for the claimed “embedded resource with a tag.” Pet., 45-48. But Lewin does not supply this missing “tag” limitation. EX2010, ¶57.

Figure 1 of the ’322 patent illustrates the claimed “embedded resource with a tag.” EX1001, FIG. 1; EX2010, ¶58.



EX1001, FIG. 1 (annotated).

As the ’322 patent explains, “the origin server is connected with a database 22, which organizes and manages stored resources in some form of storage device

(or devices), such as a storage appliance, that includes the HTML document 20 with an embedded resource 24.” EX1001, 4:22-26. “In this example, the embedded resource 24A is a link to some form of content that the user can obtain by selecting the link when the Web page is displayed.” *Id.*, 4:28-31; EX2010, ¶59.

The ’322 patent further explains that “[t]he embedded resource may include **a tag indicating that a resource address compute engine 30 should be contacted** via a call to an API 26.” EX1001, 4:32-34 (emphasis added). In an example, “the tag is ‘\$’ which causes the origin server 18 to call the API 26 to retrieve the address (only) or the complete URL,” and “[h]ence, **the presence of the tagged resource 24A in the HTML document 20 at the origin servers causes the origin server to call the API, the result of which causes the HTML document returned to the requesting client to include a modified embedded resource 24B.**” *Id.*, 4:40-48 (emphasis added); EX2010, ¶60.

This operation is expressly recited in claim 1, requiring “at least one embedded resource with a tag,” and “**upon identification of the tag**, using at least one delivery parameter to **obtain, by the origin system, a modified embedded resource** from a content delivery network that is separate from the origin system.” EX1001, 10:6-12. In other words, the claim 1 requires that the recited “tag” be used to initiate the operation of obtaining “the modified embedded resource [that]

includes a direct link to a node within the content delivery network that can deliver content associated with the embedded resource.” *Id.*, 10:6-18; EX2010, ¶61.

Petitioner relies on incorporation of Lewin’s tags within Verma’s system “to have a uniform and efficient way to identify URLs,” without modification to the function they serve in Lewin’s system. Pet., 49-50. But Lewin’s purported “tag,” in contrast to the “tag” recited in claim 1 of the ’322 patent, is not used to obtain a “modified embedded resource [that] includes a direct link to a node within the content delivery network that can deliver content associated with the embedded resource,” as required by claim 1. EX2010, ¶¶62-63. In particular, the Petition alleges that in Lewin, “‘directives [] are placed in HTML pages’ and ‘evaluated on a server before the page is served,’ which ‘enable[s] the server to add dynamically-generated content to an existing HTML page.’” Pet., 47 (citing EX1007, ¶28). According to Petitioner, “[t]he ‘dynamic assembly instructions’ (URLs (embedded resources)) are ‘specif[ied]’ by ‘tags,’ like a ‘<esi:include> tag.’” Pet., 47 (citing EX1007, ¶¶28, 30, 49).

Petitioner, however, does not explain how Lewin’s “<esi:include>” tags are used (alone or in combination with Verma) to “obtain ... a modified embedded resource” that “includes a direct link to a node within the content delivery network that can deliver content associated with the embedded resource,” as required by claim 1. *See* Pet., 46-50, 54-56; EX2010, ¶63. Rather, as Petitioner notes, Lewin’s

tags instruct the server to *add* “content to an existing HTML page,” not a new *link to the content*, as the claims require. Pet., 47 (citing EX1007, ¶28); EX2010, ¶63. Indeed, the Petition merely alleges that a “POSITA would have understood Lewin’s tags allow for identification of URLs, and would have wanted to incorporate tags in Verma to have a uniform and efficient way to identify URLs,” and that such tags would “enable Verma’s system to utilize and identify advantageous embedded content by providing a solution for dynamic identification and execution of the URLs.” Pet., 50, 55. Such analysis fails to address the claim language, requiring obtaining a “modified embedded resource [that] includes a direct link to a node within the content delivery network that can deliver content associated with the embedded resource” upon “identification of the tag.” EX1001, 10:6-18; EX2010, ¶64.

Moreover, Petitioner provides no reason why a POSITA would have sought to modify Verma’s existing markup scheme. EX2010, ¶65. As Petitioner acknowledges, Verma already includes standard HTML tags (e.g., “a href” tags) to identify URLs. Pet., 51. The Petition also does not explain how replacement of these tags with those disclosed by Lewin would improve or alter Verma’s operation, or how they would function with Verma’s existing content broker and cache servers. EX2010, ¶65; Pet., 46-50, 54-56. Further, by missing *how* and *why* a POSITA would look to Lewin’s teachings, the Petition also fails to show an

expectation of success of using Lewin’s “<esi:include>” tags to replace Verma’s standard HTML tags, such that the combination with Lewin’s tag would meet the claimed “obtain ... a modified embedded resource” that “includes a direct link to a node within the content delivery network that can deliver content associated with the embedded resource.” EX2010, ¶65.

In the end, even had Lewin supplied the claim elements missing from Verma (it does not), the mere existence of those elements in the prior art is insufficient to render obvious claim 1. *Virtek Vision Int'l ULC v. Assembly Guidance Sys., Inc.*, 97 F.4th 882, 887 (Fed. Cir. 2024) (“[t]he mere fact that these possible arrangements existed in the prior art does not provide a reason that a skilled artisan would have substituted” one arrangement for the other).

As Lewin and Verma are not obvious to combine, and even if combined, still fail to teach “obtain[ing] ... a modified embedded resource” that “includes a direct link to a node within the content delivery network that can deliver content associated with the embedded resource,” the Petition fails to establish a reasonable likelihood of prevailing with respect to independent claim 1. EX2010, ¶¶63-66.

VI. CONCLUSION

For at least the foregoing reasons, Petitioner has failed to show a reasonable likelihood of prevailing with respect to any challenged claim. Therefore, the Board should deny institution.

Respectfully submitted,

STERNE, KESSLER, GOLDSTEIN & FOX PLLC

/Jason D. Eisenberg/

Jason Eisenberg, Reg. No. 43,447
Daniel S. Block (Reg. No. 68,395)
Kristina Caggiano Kelly (Reg. No. 58,164)
Steven M. Pappas (Reg. No. 73,904)
Counsel for Patent Owner

Date: September 16, 2025

1101 K Street, NW, 10th Floor
Washington, DC 20005
(202) 371-2600

CERTIFICATE OF WORD COUNT (37 C.F.R. § 42.24(d))

1. This Patent Owner Preliminary Response complies with the type-volume limitation of 14,000 words, comprising 6,034 words, excluding the parts exempted by 37 C.F.R. § 42.24(a)(1).

2. This Patent Owner Preliminary Response complies with the general format requirements of 37 C.F.R. § 42.6(a) and has been prepared using Microsoft® Word 2016 in 14-point Times New Roman font.

Respectfully submitted,

STERNE, KESSLER, GOLDSTEIN & FOX PLLC

/Jason D. Eisenberg/

Jason D. Eisenberg
Registration No. 43,447
Counsel for Patent Owner

Date: September 16, 2025

1101 K Street, NW, 10th Floor
Washington, DC 20005
(202) 371-2600

CERTIFICATE OF SERVICE (37 C.F.R. § 42.6(e))

I certify that the above-captioned **PATENT OWNER PRELIMINARY RESPONSE UNDER 37 C.F.R. § 42.107(a)** and associated Exhibits 2010-2014 were served in their entireties on September 16, 2025, upon the following parties via electronic mail:

Erika H. Arner (Lead Counsel) erika.arner@finnegan.com
Kara A. Specht (Back-Up Counsel) kara.specht@finnegan.com
Sydney R. Kestle (Back-Up Counsel) sydney.kestle@finnegan.com
Cara E. Regan (Back-Up Counsel) cara.regan@finnegan.com
Milan F. Feliciello (Back-Up Counsel) milan.feliciello@finnegan.com
Daniel C. Tucker (Back-Up Counsel) daniel.tucker@finnegan.com
Google-Sandpiper-IPRs@finnegan.com
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, LLP

Respectfully submitted,

STERNE, KESSLER, GOLDSTEIN & FOX PLLC

/Jason D. Eisenberg/

Jason D. Eisenberg
Registration No. 43,447
Counsel for Patent Owner

Date: September 16, 2025

1101 K Street, NW, 10th Floor
Washington, DC 20005
(202) 371-2600