

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner

v.

SANDPIPER CDN, LLC,
Patent Owner

Case IPR2026-00180
U.S. Patent No. 10,701,173

**DECLARATION OF SAMUEL H. RUSS, PH.D.,
IN SUPPORT OF PATENT OWNER'S PRELIMINARY RESPONSE**

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

Sandpiper EX2008
Microsoft v. Sandpiper
IPR2026-00180

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MY BACKGROUND AND QUALIFICATIONS	2
III.	LIST OF DOCUMENTS CONSIDERED	5
IV.	LEGAL UNDERSTANDING.....	6
	A. My Understanding of Claim Construction.....	6
	B. My Understanding of Obviousness.....	7
	C. My Understanding of a Person Having Ordinary Skill in the Art	9
V.	PERSON OF ORDINARY SKILL IN THE ART	10
VI.	OVERVIEW OF THE STATE OF THE ART.....	10
VII.	THE '173 PATENT.....	12
	A. Overview of the '173 Patent.....	12
	B. Overview of the Prosecution History of the '173 Patent	13
VIII.	OVERVIEW OF THE PRIMARY REFERENCES	15
	A. O'Rourke.....	15
	B. Middleton	17
IX.	THE PROPOSED OBVIOUSNESS GROUNDS.....	18
	A. Ground 1	18
	B. Ground 2.....	24
X.	CONCLUSION.....	29

I, Samuel H. Russ, hereby declare as follows.

I. INTRODUCTION

1. I have been retained as an expert witness on behalf of Patent Owner Sandpiper CDN, LLC (“Patent Owner”) for the above-captioned *inter partes* review (“IPR”). I am being compensated for my time in connection with this IPR at my normal hourly rate. My compensation is not contingent on either my expert opinion or the outcome of this litigation. I have no other interest in this proceeding.

2. I understand that this declaration accompanies Patent Owner’s Preliminary Response filed in this IPR involving U.S. Patent No. 10,701,173 (“the ’173 patent”) (EX1001), which resulted from U.S. Application No. 15/588,935 (“the ’935 application”), filed May 8, 2017, which claims priority to U.S. Application No. 14/570,512 (“the ’512 application”), filed December 15, 2014. My opinions regarding the knowledge in the art at the time of the ’173 patent are based on this date of December 15, 2014. Although I express no opinion as to whether the ’173 patent is entitled to the priority date of the ’512 application, my opinions regarding the knowledge in the art would not materially differ if based on the date of May 8, 2017.

3. In preparing this declaration, I have reviewed the ’173 patent and each of the documents cited herein, in light of general knowledge in the art before the time of the ’173 patent. In formulating my opinions, I have relied upon my

experience, education, and knowledge in the relevant art. In formulating my opinions, I have also considered the viewpoint of a person of ordinary skill in the art (“POSITA”) (as discussed further below in Section V) at the time of the ’173 patent.

II. MY BACKGROUND AND QUALIFICATIONS

4. I received a Bachelor’s degree in Electrical Engineering from the Georgia Institute of Technology (“Georgia Tech”) in 1986 and a Ph.D. in Electrical Engineering from Georgia Tech in 1991.

5. From 2007 to the present, I have been a member of the faculty of the University of South Alabama as an Assistant and Associate Professor in the Department of Electrical and Computer Engineering. During that time, I have won awards for excellent teaching and have been actively publishing research in home networking and digital video recording (DVR) technologies. I am active in the Institute of Electrical and Electronic Engineers (IEEE) and was a Distinguished Lecturer for the IEEE Consumer Electronics Society. As an example of the type of work I perform at South Alabama, I managed the design, construction, and launch of a fully functional satellite from the International Space Station into space in 2022.

6. From 2000 to 2007, I worked for Scientific-Atlanta (now Cisco’s Service Provider Video Tech. Group), where I managed a cable set-top box (STB)

design group that designed four STB models, including the Explorer 4200 (non-DVR) and 8300 (DVR) models. Both models sold several million units. As design-group manager, I was responsible for managing the design and prototyping activities of the group and for interfacing with other groups (especially integrated-circuit design, procurement, software developers, the factory where prototypes were built, and product managers) and for maintaining the hardware and mechanical development schedule. Since the products were produced in extremely high volumes, the projects had very high visibility in the company, and therefore carried a great deal of responsibility.

7. Also while at Scientific-Atlanta, I became a staff expert in home networking, conducting demonstrations of wireless video technology and managing a group that developed a new coaxial home networking system. The coaxial system won a Technology and Engineering Emmy® Award in 2013. I became a staff expert in DVR reliability, and led a team that improved the software, hardware, repair, and manufacturing processes. I am a named inventor on fifty-one (51) patent applications that were filed while I was at Scientific-Atlanta, twenty eight (28) of which have issued as U.S. patents as of the writing of this report.

8. Both my work with home networking and coaxial DVR video distribution were technologies that improve content distribution networks (CDNs).

In order to implement these systems, I had to develop a detailed understanding of, *e.g.*, bandwidth and resource management, video distribution, and the requirements for storage media to support multiple video streams. For example, because of my work on multi-room DVR and hard drive reliability, I identified the level of performance needed for a hard disk drive to support seven simultaneous streams of video when functioning as a home video server. The set-top boxes I designed supported video on demand and switched digital video, both forms of content distribution.

9. From 1999 to 2000, I was a Staff Electrical Engineer and then Matrix Manager at IVI Checkmate (now Ingenico), where I managed the hardware design team that completed the design of the eN-Touch 1000 payment terminal. This terminal was in widespread use, for example, at the self-checkout at Home Depot.

10. I also served on the faculty of Mississippi State University from 1994 to 1999 as an Assistant Professor in the Department of Electrical & Computer Engineering where I taught circuit board design and two-way interactive video classes, among other things.

11. I have also authored 32 journal articles and conference papers. A conference paper on digital video recording won second place in a “best paper” competition at the 2011 International Conference on Consumer Electronics in Las Vegas, NV.

12. I understand that a copy of my complete curriculum vitae has been submitted as Exhibit 2011, which includes a list of my publications and the cases in which I have testified as an expert in deposition or trial in the past four years, and other details regarding my expertise and qualifications.

III. LIST OF DOCUMENTS CONSIDERED

13. I have considered the following documents listed below in addition to any other documents cited in this declaration. To the best of my knowledge, these documents are true and accurate copies of what they purport to be. An expert in the field would reasonably rely on them to formulate opinions such as those set forth in this declaration.

Paper No.	Description
1	Petition for <i>Inter Partes</i> Review of U.S. Patent No. 10,701,173

Exhibit No.	Description
1001	U.S. Patent No. 10,701,173 (“the ’173 Patent”)
1002	Prosecution History of the ’173 patent
1003	Declaration of Mark R. Lanning
1004	<i>Curriculum Vitae</i> of Mark R. Lanning
1005	U.S. Patent No. 7,912,921 (“O’Rourke”)
1006	U.S. Patent Publication No. 2011/0060812 (“Middleton”)
1007	U.S. Patent Publication No. 2012/0096106 (“Blumofe”)

Exhibit No.	Description
1008	[MS-MSSO]: Media Streaming Server System Overview (Microsoft Corporation, 2009), available at https://web.archive.org/web/20100130235412/http://download.microsoft.com/download/9/5/e/95ef66af-9026-4bb0-a41da4f81802d92c/Windows_communication_protocols.zip (“MS-MSSO”)
2009	<i>Curriculum Vitae</i> of Samuel H. Russ, Ph.D.
2010	“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)
2011	“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)
2012	Basu, S., et al., “Adaptive TTL-Based Caching for Content Delivery,” <i>IEEE/ACM Transactions on Networking</i> , 99:1-15 (April 2018)

IV. LEGAL UNDERSTANDING

A. My Understanding of Claim Construction

14. I understand that during an *inter partes* review proceeding, claims are to be construed in light of the specification as would be read by a person of ordinary skill in the relevant art at the time of the invention. I understand that claim terms are given their plain and ordinary meaning as would be understood by a person of ordinary skill in the relevant art in the context of the patent’s specification and prosecution history. A claim term, however, will not receive its

ordinary meaning if the patentee acted as his or her own lexicographer and clearly set forth a definition of the claim term in the specification. In this case, the claim term will receive the definition set forth in the patent.

B. My Understanding of Obviousness

15. I understand that a patent claim is invalid if the claimed invention would have been obvious to a person of ordinary skill in the art at the time the application was filed. I understand that this means that even if all of the requirements of the claim cannot be found in a single prior-art reference that would anticipate the claim, the claim can still be invalid.

16. To obtain a patent, a claimed invention must have been, as of its priority date, nonobvious in view of the prior art in the field. I understand that a patent claim is obvious when the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art.

17. I understand that to prove that prior art or a combination of prior art renders a patent claim obvious, it is necessary to: (1) identify the particular references that, singly or in combination, render the patent claim obvious; (2) specifically identify which elements of the patent claim appear in each of the

asserted references; and (3) explain how the prior art references could have been combined in order to create the inventions claimed in the patent claim.

18. I also understand that when considering the obviousness of a patent claim, one should consider whether a teaching, suggestion, or motivation to combine the references exists so as to avoid impermissibly applying hindsight when considering the prior art. I understand this test should not be rigidly applied, but that the test can be important to avoiding such hindsight.

19. I understand that certain objective indicia can be important evidence as to whether a patent is obvious or nonobvious. Such indicia include:

(1) commercial success of products covered by the patent claims; (2) a long-felt need for the invention; (3) failed attempts by others to make the invention; (4) copying of the invention by others in the field; (5) unexpected results achieved by the invention as compared to the closest prior art; (6) praise of the invention by the infringer or others in the field; (7) the taking of licenses under the patent by others; (8) expressions of surprise by experts and those skilled in the art at the making of the invention; and (9) the patentee proceeded contrary to the accepted wisdom of the prior art.

20. I also understand that “obviousness” is a legal conclusion based on the underlying factual issues of the scope and content of the prior art, the differences between the claimed invention and the prior art, the level of ordinary skill in the

pertinent art, and any objective indicia of non-obviousness. For that reason, I am not rendering a legal opinion on the ultimate legal question of obviousness. Rather, my testimony addresses the underlying facts and factual analysis that would support a legal conclusion of obviousness or non-obviousness, and when I use the term obvious, I am referring to the perspective of one of ordinary skill at the time of the '173 patent.

C. My Understanding of a Person Having Ordinary Skill in the Art

21. I understand that a POSITA is presumed to be aware of all pertinent art, thinks along conventional wisdom in the art, and is a person of ordinary creativity.

22. I understand that the following should be considered when deciding the level of ordinary skill in the art that someone would have had at the time the claimed invention was made:

- the levels of education and experience of persons working in the field;
- the types of problems encountered in the field; and
- the sophistication of the technology.

23. My opinion below explains how a POSITA would have understood the technology described in the documents I have identified herein around the time of the '173 patent.

24. Regardless of whether I use “I” or “POSITA” during my technical analysis below, all of my statements and opinions are always to be understood to be based on how a POSITA would have understood or read a document at the time of the ’173 patent.

V. PERSON OF ORDINARY SKILL IN THE ART

25. For purposes of this declaration, I have applied Petitioner’s articulation of a POSITA. EX1003, ¶94. It is my opinion that the definition of a POSITA would not materially affect the limited analysis required for deciding institution and presented in this declaration. If called to give further opinions in this proceeding, I reserve the right to reconsider the definition of a POSITA and arrive at my own conclusion.

VI. OVERVIEW OF THE STATE OF THE ART

26. Content delivery networks (CDNs) provide critical services that enable content providers to quickly deliver online content to millions of consumers simultaneously over the Internet. 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, 2.

27. 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, 2-3.

28. The availability of CDN servers around the world allowed service providers to 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. EX2010, 3-6; EX2011, 2-3. This service and its architecture were quickly adopted by many others in the industry. EX2010, 3-6; EX2011, 2-3.

29. In the CDN model, duplicate copies of content are stored in servers located over a wide geographic area (*e.g.*, across the country or around the world). EX2010, 2; EX2011, 2-3. Propagating data from origin servers to edge servers is known as "caching." EX2010, 3-6; EX2011, 2-3. In other words, when data is propagated from its origin to one of the geographically distributed CDN servers, the data is cached at the CDN server. EX2010, 3-6; EX2011, 2-3. Using a CDN, end-users connect to an optimal edge server—one that that is closer to them or that has available capacity. EX2011, 2-3. Distributing content across a network of

servers alleviated data congestion issues, and allowing consumers to connect to edge servers located near them reduced latency. Importantly, the CDN model is invisible to the user—they simply see the content they requested. Note that the closest server may not necessarily refer to one that is geographically closest, but rather has the lowest latency. EX2010, 3; EX2011, 1.

30. In the early-to-mid 2000s, demand for CDNs exploded. EX2010, 7-8. Content providers like Google and YouTube use CDN technology to meet their enormous data streaming needs, as well as for video search results. EX2001, 1-3. As a result, techniques for controlling how users access content on a CDN became increasingly important.

VII. THE '173 PATENT

A. Overview of the '173 Patent

31. The '173 patent is entitled “Caching in a Content Delivery Framework,” and was filed as U.S. Application No. 15/588,935 on May 8, 2017. It is a continuation of U.S. Patent No. 9,648,127, filed on December 15, 2014. The '173 patent is directed to improvements in CDN technology, specifically addressing how CDNs manage caching policies, prevent stale content from being served, and handle subscriber-specific caching controls. CDN caches store copies of subscriber content so that end-user requests can be served by CDN nodes.

32. Embodiments of the '173 patent address technical issues with handling requests and timely providing content. The '173 patent describes enhancing this process by dynamically determining the appropriate caching policy at the time a request for content is received, enabling the CDN to honor up-to-date rules while improving efficiency and flexibility. The inventors pioneered the described unconventional technical solutions for maintaining content freshness and enforcing subscriber-specific caching directives in a scalable, low-overhead manner to address problems associated with stale content, inconsistent policy application, and the operational burden of widespread invalidations.

B. Overview of the Prosecution History of the '173 Patent

33. I understand that the examiner of the '173 patent included consideration of the primary reference used in Ground 1 in this Petition (O'Rourke). Via Information Disclosure Statements, Patent Owner cited O'Rourke, the published application of O'Rourke ("pub-O'Rourke"), and an International Preliminary Report on Patentability (IPRP) that applied pub-O'Rourke to Patent Owner's then-pending claims (on anticipatory grounds). EX1002, 278-284, 304. Along with issuance of the first Office Action, the Examiner indicated they considered O'Rourke, pub-O'Rourke, and the IPRP. EX1002, 272, 277.

34. The Examiner, after being presented with the IPRP containing anticipatory arguments based on pub-O'Rourke, chose to issue a different anticipation rejection using a different reference (U.S. Patent No. 9,984,166), which was found to be closer prior art than O'Rourke. EX1002, 262-270. A nonstatutory double patenting (NDP) rejection with respect to the issued parent was also issued. *Id.* Patent Owner responded with rebuttal arguments. EX1002, 250-258.

35. Thereafter, a series of Office Actions and responses ensued, involving an obviousness rejection (relying on a combination of U.S. Patent No. 9,984,166 and U.S. Publication No. 2015/0347248), an Examiner interview, Request for Continued Examination, Patent Owner's submission of a terminal disclaimer, and amendments to the claims. EX1002, 104, 128-136, 139, 142-155, 160-168, 171-182, 186-192, 208-220, 232, 238-247.

36. Amendments included the addition of "after determining that the content is cached locally" to step (B) and "wherein determining the current cache policy comprises determining a time period for expiration of content in the cache" to step (B)(1) of the independent claims. EX1002, 112-114. These amendments successfully distinguished the closest prior art, and they distinguish both primary references cited in the present Petition. After these amendments, the Examiner

entered a Notice of Allowance. EX1002, 18-25. Patent Owner then paid the issue fee and the patent issued.

VIII. OVERVIEW OF THE PRIMARY REFERENCES

37. I understand that the Petition relies on different combinations of four references across the various Grounds. I also understand that the independent claims, which are my focus in this Declaration, only rely on two references, singularly, to allege obviousness. I provide brief overviews of these two primary references below.

A. O'Rourke

38. O'Rourke discloses a method and apparatus for selecting cache and proxy policy in a cache server used to distribute media content in a content delivery network. EX1005, Abstract, 1:13-15. Figure 5 illustrates a procedure for handling a client request for pre-recorded content.

39. In O'Rourke's procedure 500, after a client requests content (block 502), the cache server first "identifies the policies related to requests for pre-recorded content" at block 504. EX1005, 6:12-14. Block 504 is keyed to the type of request and identifies which pre-defined cache policies may apply to handling that request. O'Rourke does not describe block 504 as being conditioned on whether the requested content is cached locally, nor as determining any policy specific to a cached object. Consistent with this structure, O'Rourke describes the

identification of “policies related to requests for pre-recorded content” as a separate step prior to determinations about whether a cached version is acceptable, treating policy identification and cache-freshness evaluation as distinct operations. EX1005, 6:12-37.

40. At block 505, the cache server “identifies whether the requested content is stored on the cache server.” EX1005, 6:14-19. This is a binary cache-presence check that determines only whether a copy of the content exists on the cache server, without reference to any cache policy or expiration rule. *Id.* O’Rourke does not describe any cache-policy determination as being conditioned on the outcome of block 505. *See* EX1005, 6:20-37.

41. If the content is stored on the cache server, O’Rourke proceeds to block 510, where the cache server determines “whether the cached version of the requested content is an acceptable version.” EX1005, 6:31-37. O’Rourke explains that this determination concerns freshness or version matching, such as whether the cached version matches the origin server’s version or falls within a permitted age window. *Id.* Block 510 is described as a version-acceptability or freshness check, not as a determination of a cache policy associated with the content. EX1005, 6:31-37, 7:64-67. The acceptability check evaluates cached content against criteria such as version matching or permitted age, rather than identifying which cache policy governs the handling of the request. *Id.*

42. Thus, O'Rourke discloses separate steps for (i) identification of policies related to a request (block 504), (ii) determination of whether content is cached (block 505), and (iii) evaluation of whether a cached copy is acceptable or fresh (block 510) as distinct steps, without describing block 510 as a determination of a current cache policy made after confirming that the content is cached locally. EX1005, 6:12-37.

B. Middleton

43. Middleton discloses a cache server architecture for streaming content in a content delivery network, including mechanisms for caching content, streaming content to clients, and revalidating cached content. EX1006, ¶¶[0033]-[0034]. In Middleton, “generally, each content item has an associated [TTL] value.” EX1006, ¶[0052]. Middleton describes a cache server that manages cached content by comparing content age to an associated time-to-live (TTL) value and reacting when cached content becomes stale. EX1006, ¶[0080], FIG. 5.

44. If the content has not exceeded its TTL, the cached content is streamed to the requester (operation 508). EX1006, ¶[0082]. When the cached content is stale, Middleton describes sending a HEAD request to a media access server to attempt revalidation (operation 510). EX1006, ¶¶[0081]-[0082]. If the media access server indicates that the cached content remains valid, the cache server updates the TTL associated with the content and continues to treat the

cached content as usable (operation 514). EX1006, ¶[0082]. If revalidation fails, the stale content is deleted and the cache server retrieves a fresh copy of the content (operations 516, 518). EX1006, ¶[0083].

45. Throughout this process, Middleton treats TTL values as pre-existing attributes associated with content and uses them as part of a freshness or staleness check. EX1006, ¶[0052]. Middleton does not describe determining or selecting a cache policy at request time. Instead, it explains that each content item has an associated TTL value that is read and compared against content age to assess staleness, with the cache server reacting based on the pre-associated TTL. EX1006, ¶[0080]. TTL refresh occurs only after a staleness event and in response to revalidation exchange, functioning as a reactive update to expiration information content rather than a determination of which cache policy governs the request. EX1006, ¶¶[0081]-[0082].

IX. THE PROPOSED OBVIOUSNESS GROUNDS

46. For purposes of this declaration, I have been asked to opine on Grounds 1 and 2 as they relate to the independent claims, specifically as described in Grounds 1A and 2A.

A. Ground 1

47. In Ground 1A, Petitioner's theory rests on the flawed recasting of O'Rourke's block 510—an "acceptable version" or freshness check—as the

claimed determination of a current cache policy, based solely on its placement later in the flow. Pet., 26-29. Petitioner asserts that because O’Rourke’s block 510 is reached only after determining that content is cached locally, the “acceptable version” decision performed at block 510 constitutes the claimed determination of a current cache policy. Pet., 26. Under Petitioner’s theory, any freshness or staleness evaluation of block 510 is treated as a cache-policy determination, disregarding the fact that O’Rourke explicitly determines cache policies *prior* to locally cached content determinations and describes block 510 *only* as a version-acceptability check. EX1005, 6:12-14, 6:31-34.

48. Petitioner’s theory is wrong because it conflates policy identification with cache-freshness evaluation and ignores O’Rourke’s explicit teachings for each of its blocks/steps. Policy identification involves determining the cache policy governing how a CDN operates. Cache-freshness evaluation is a separate process that may or may not apply a cache policy, and evaluates cached content based on that policy.

49. Petitioner’s reliance on O’Rourke’s block 510 is erroneous in two respects. *First*, Petitioner’s reading of the claimed “when it is determined by the node that said content is cached locally, then, after determining that the content is cached locally: (B)(1) determining, by the node, a current cache policy associated with the content,” is overbroad. Petitioner effectively reads out the order of steps

and cause-and-effect recited through the words “when” and “then, after.” *Second*, even with Petitioner’s overbroad reading of the claims, O’Rourke’s block 510 still does not teach the cache policy.

50. Block 510 determines whether the cached version of the requested content is an “acceptable version” (e.g., whether it matches the origin server’s version or falls within a freshness window). EX1005, 6:34-37. Block 510 is a version/freshness assessment, not a determination of a content policy. That distinction matters because O’Rourke’s “cache policies” refer to a broader set of operational behaviors (e.g., “distributing media content, handling cache misses, and prefetching media content”) that govern how a CDN operates. EX1005, 2:60-62.

51. Even if the claimed “determining a current cache policy associated with requested content” is interpreted broadly to include any identification or determination of a policy relevant to the request, O’Rourke places that concept squarely at block 504, not block 510.

52. O’Rourke is clear about what blocks 504 and 510 perform. O’Rourke’s “cache policies” include “distributing media content, handling cache misses, and prefetching media content.” EX1005, 2:60-62. Such cache policies are identified at block 504. EX1005, 6:12-14. Block 510, by contrast, answers a narrower question: whether cached content remains fresh enough to serve (not

which cache policy governs the request). O'Rourke instead teaches identification of relevant policies at block 504 ("identifies the policies related to requests for pre-recorded content"). EX1005, 6:12-14.

53. Petitioner's argument tries to impute the policy inquiry of block 504 into the freshness check of block 510. That move trades on a superficial similarity in terminology. While the '173 patent's cache policies govern how a *cached* object is treated at request time, O'Rourke's block 510 merely asks "is this cached content fresh enough" to serve. This type of evaluation already existed in conventional HTTP, which the '173 patent contemplates with mention of evaluations like TTL. EX1001, 5:55-6:3. It does not teach or suggest the specific cache policy of the challenged claims.

54. Thus, Petitioner's recasting of block 510 as "determining a current cache policy" is inconsistent with O'Rourke's description of the functions at blocks 504 and 510. O'Rourke does not teach or suggest that its cache policy identification step (block 504) occurs "after determining that the content is cached locally." And block 510 does not perform the policy determination Petitioner imputes to it.

55. Petitioner attempts to avoid the clear discrepancy between the challenged claims and O'Rourke using alternative theories that redefine which of O'Rourke's blocks supposedly perform the claimed determination. Pet., 12, 22-23.

But Petitioner’s stated rationale for rearranging the logic and flow of O’Rourke—that neither function depends on the other—is (1) incorrect, and (2) would not provide a reason to substitute one for the other even if it were true.

56. For instance, Petitioner’s first theory conflating and switching the functionalities of blocks 504 and 510 fails as discussed above:

[O’Rourke’s] procedure 500 transitions at line 501A from a prior identification of policies related to requests for pre-recorded content (504) to application of cache policies for the requested content (e.g., block 510). ... Thus, after line 501A (which occurs after determining that the content is cached locally as recited in 1[b]/8[b]) *O’Rourke determin[es]...a current cache policy (to determine whether a cached version of the content is an acceptable version at block 510).*

Pet., 28 (emphasis added).

57. Petitioner then turns to reordering O’Rourke’s disclosed steps. In the second theory, Petitioner concedes that O’Rourke’s blocks 504 and 505 “do not depend on” one another, asserting that:

[i]t would have been obvious to, in response to a request for pre-recorded content, identify whether the requested content is stored on the cache server (505) and then identify the policies related to requests for pre-recorded content (504), *because neither function depends on the other function.*

Pet., 12 (emphasis added).

58. That concession is dispositive. It confirms that O’Rourke does not satisfy the claimed dependency requiring that, “when it is determined [] that said content is cached locally, then, after determining that the content is cached locally,” the node determines a current cache policy. In O’Rourke, block 504 is keyed to requests (“cache server identifies the policies related to request for pre-recorded content”) and block 505 is keyed to whether cached content is present (“cache server identifies whether the requested content is stored on the cache server”). EX1005, 6:12-14. Petitioner’s own premise establishes that O’Rourke’s block 504 is not conditioned on a prior determination that the content is cached locally—it undercuts the conditional dependencies and sequencing recited in the independent claims.

59. And even if O’Rourke’s blocks 504 and 505 could be interchanged (which Patent Owner does not concede), the claims still require that the subsequent determination of the expiration period be tied to the determination that the content is cached locally, not merely temporally sequenced after it. *See* claims 1 and 8 (“wherein determining the current cache policy comprises determining a time period for *expiration of content in the cache*”). Petitioner’s reordering may supply a temporal “after,” but it does not create the required causal link—namely a determination of an expiration period made because the content was found cached

locally. *See* claims 1 and 8 (“when it is *determined* by the node that said *content is cached locally*, then, *after determining that the content is cached locally*”).

60. O’Rourke’s block 504 is never described as determining an expiration period based on content having been found in the cache. That is, merely performing block 505 first does not transform block 504 into a determination that uses the “cached locally” result to determine an expiration for content in “the cache.” By conceding that blocks 504 and 505 “do not depend on each other,” Petitioner forecloses any conditional linkage.

61. In summary, Petitioner’s fallback theory defeats itself. By asserting that O’Rourke’s blocks 504 and 505 “do not depend on each other,” Petitioner confirms that O’Rourke does not disclose the conditional linkage the claims require—namely, that after the node determines content is cached locally, the node then determines a current cache policy based on that determination. The conceded independence between blocks 504 and 505 forecloses any suggestion that O’Rourke conditions cache-policy determination on cache presence, and no reordering of those steps can supply the dependencies the claims require.

B. Ground 2

62. Ground 2A fails at least because Middleton does not disclose or suggest the claimed “determining [] a current cache policy.” Middleton fails to teach or suggest this limitation for a related, but even more straightforward reason

than O'Rourke. Rather than identifying any determination of a cache policy in Middleton, Petitioner collapses Middleton's time-to-live (TTL) staleness check into a policy determination. But a TTL staleness check and a policy determination are two different procedures that serve two different functions and follow two different logical structures. Petitioner misapprehends the technology.

63. Petitioner's entire theory is that Middleton's "content's TTL value *is a cache policy* associated with the content" and that comparing "the age of the locally cached content to a TTL value associated with the content" satisfies the determining claimed step. Pet., 67; EX1006, ¶[0080]. A TTL value is not a cache policy—not even as Middleton describes it. Petitioner is simply wrong to equate these things.

64. Critically, Middleton does not describe determining which cache policy is "current" (the claims require a determination of a "*current* cache policy") because Middleton has no mechanism and no need for making that determination. Middleton operates in a fundamentally different way: by relying on a pre-existing TTL associated with cached content. EX1006, ¶[0052] ("Generally, each content item has an associated time-to-live (TTL) value."). Middleton uses the pre-existing TTL as an input to a staleness check. *Id.* ("check the age of the content to determine if the content is stale"). That staleness check applies a pre-associated expiration value that is already bound to the cached content; Middleton does not

describe determining, at request time, which expiration directive or rule is currently operative for the content.

65. The '173 patent identifies a different architectural problem that arises when expiration behavior can change after content is cached—namely, “invalidat[ing] cached content ... every time a caching policy for that content is set,” an approach that is potentially expensive. EX1001, 3:62-4:3. Consistent with this framing, technical literature on TTL-based caching systems—including work focused on dynamically adapting TTL values—treats TTL as a parameter, rather than as a mechanism for determining which cache policy applies at request time. EX2012, 1 (“a TTL parameter”). The claimed determining step addresses the problem by determining the current cache policy at request time, ensuring that the latest cache policy applies without requiring forced invalidations when an update is made to the cache policy. Middleton does not disclose any request-time mechanism for determining whether updated caching rules govern cached content and instead describes applying TTL values that are already associated with the cached content. EX1006, ¶[0052].

66. Even if a TTL value were treated as a “policy” or part of one (which Patent Owner does not concede), Middleton still never discloses determining that policy’s *current* status. It does not even contemplate the kind of policy changes that would make such a status something to be determined. Middleton functions

entirely by reading whatever TTL is already associated with the content and checking whether the content has expired. But such a comparison against an assumed, static expiration value is not a determination of a “current cache policy” in a dynamic system.

67. Petitioner’s theory again trades on a superficial similarity in terminology rather than substantive disclosure. In Middleton, a TTL is simply expiration metadata associated with content and used to assess staleness. EX1006, ¶¶[0052], [0080]. It is not identified, selected, or determined by the node as a governing cache policy at request time. By contrast, the ’173 patent’s cache policies are rules that govern how cached content is handled when a request is received, including determining an expiration period as part of that policy. *See* EX1001, 5:55-6:3. Consistent with that distinction, even research directed to improving TTL-based caching through dynamic or adaptive techniques treats TTL as an expiration-control parameter and does not describe TTL mechanisms as determining, at request time, which caching policies govern how a particular request is handled. EX2012, 1 (“a TTL parameter”).

68. Treating Middleton’s TTL comparison as a policy determination collapses, and thereby erases, a legally meaningful distinction that the claim requires. It reads out an expressly recited order of steps, and an expressly recited cause-and-effect relationship between two operations. In doing so, Petitioner

improperly substitutes Middleton's routine freshness check for the claimed node's affirmative determination of a current cache policy.

69. Put another way, Middleton's process contains no inquiry into whether the TTL in use reflects the policy currently in effect at the node, nor any selection among policies. Middleton's comparison succeeds regardless, without ever ascertaining whether the policy is current. Nor does Middleton's revalidation or TTL refresh cure this deficiency. Middleton updates TTL values only after a staleness event and only reactively, in response to "a HEAD request [] sent to a media access server (MAS)." EX1006, ¶[0082]. That reactive update mechanism does not involve determining what cache policy is currently in effect at the node; it simply replaces one expiration value with another after the system has already determined that content is stale.

70. In summary, Middleton describes a staleness-checking and reactive update mechanism, not a determination of a current cache policy. The node never determines which caching policy governs the content before making its serve decision; it merely compares content age to a pre-existing expiration value and proceeds accordingly. That is insufficient under any reasonable construction of "determining, by the node, a current cache policy."

X. CONCLUSION

In signing this declaration, I recognize that the declaration will be filed as evidence in an *inter partes* review before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required, I will appear for cross-examination within the United States during the time allotted.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed on April 2nd, 2026 in Mobile, AL.



Samuel H. Russ, Ph.D.