

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC, CELLS CO PARTNERSHIP D/B/A VERIZON WIRELESS,
VERIZON CORPORATE SERVICES GROUP INC.,
T-MOBILE USA, INC., SPRINT LLC F/K/A SPRINT CORP.,
AND AT&T SERVICES, INC.,
Petitioner,

v.

HEADWATER RESEARCH LLC,
Patent Owner.

IPR2024-00944
Patent 8,589,541

**DECLARATION OF ERIK DE LA IGLESIA IN SUPPORT OF PATENT
OWNER'S PRELIMINARY RESPONSE TO THE PETITION**

I 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.

Executed on September 17, 2024, at Mountain View, California.



Erik de la Iglesia

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Background and Qualifications	2
III.	Legal Principles	6
A.	Claim Construction	6
B.	Burden of Proof	6
C.	Anticipation	6
D.	Obviousness	7
IV.	Person of Ordinary Skill in the Art.....	8
V.	Claim Construction.....	9
VI.	Dr. Wolfe’s challenge to claim 79 is unsupported.	9
A.	A POSITA would have understood that Rao’s filter need not send a request to agent 326 in order to retrieve the priorities.....	11
B.	Dr. Wolfe provides no evidence to the contrary.....	15
VII.	Dr. Wolfe’s claim 83 challenge fails for at least the same reasons as his claim 79 challenge.....	19

I, Erik de la Iglesia, hereby declare as follows:

I. Introduction

1. I am over the age of eighteen (18) years and otherwise competent to make this declaration.

2. I have been retained as an expert witness on behalf of Patent Owner for the above-captioned *inter partes* review (“IPR”). I understand that the petition for *inter partes* review involves U.S. Patent No. 8,589,541 (“the ’541 patent”).

3. I make this declaration based on my personal knowledge, educational background and training, consideration of the materials I discuss herein, and my expert opinions.

4. My work on this matter is being billed at my normal hourly consulting rate of \$750, with reimbursement for actual expenses. My compensation is not related to the outcome of any proceeding involving the ’541 patent. I have no personal interest in the outcome of the case.

5. In preparing this Declaration, I have reviewed and considered the ’541 patent, the ’541 patent’s prosecution history, the Petition and prior art references submitted with the Petition, the declaration submitted by Dr. Wolfe in this proceeding, and each document cited in my declaration.

II. Background and Qualifications

6. My qualifications for forming the opinions given in this declaration are summarized here and are addressed more fully in my curriculum vitae, which is submitted as Exhibit 2002.

7. I hold a Bachelor of Science degree in Electrical Engineering from the University of Florida, and a Master of Science degree in Electrical Engineering from Stanford University. While at Stanford, I was an NSF Graduate Research Fellow. My graduate focus included VLSI CMOS architecture, high-speed circuit design, computer networks and protocols, and artificial intelligence. I have worked in several computer and electrical engineering related fields for over 25 years resulting in 68 issued U.S. patents that have been cited over 2850 times in USPTO patent applications. Since 2003, I have served in a number of chief architect and founder roles of companies that have been acquired or gone public. My experiences include the design or implementation of networks and network-based computation and protocol systems similar to the technology of the '541 patent, including systems for content targeting based on user behavior across devices.

8. Between December 1997 and November 1999, I was a circuit design engineer at Intel Corporation within the Mobile and Handheld Products Group. My responsibilities included working on several aspects of mobile processors and chipsets including clock and power distribution, dynamic clock frequency

selection, and the circuit design, device simulation, and validation of such features. Until March of 2000, I worked on similar functionality within the Microprocessor Products Group for the Itanium 3 and 4 products.

9. My work in power distribution included thermal modeling and predictive control of silicon devices and their packaging such as multi-tiered cooling systems for embedded, mobile, and desktop environments. Such systems included multiple tiers of thermal sensors, active and passive thermal control systems, and the feedback and analytic firmware and software control. My models were part of the development of the Thermal Design Power (TDP) metrics used by OEM and system designers to plan thermal control solutions for consumer and industrial products.

10. Between March 2000 and September 2003, I worked as a logic designer and architect for WebStacks and, through acquisition, Extreme Networks. WebStacks built a fully hardware-based TCP and HTTP processing stack for network processing functions including load balancing, proxy processing, and content rewriting. I was responsible for the architecture and design of the HTTP processing engine including header extraction and processing.

11. Between August 2003 and August 2007, I worked as Founder, Chief Architect and Director of Engineering for Reconnex, later acquired by McAfee. Reconnex built a gigabit line-rate network security analyzer capable of

classification, heuristic analysis, and policy enforcement on arbitrary protocols and content types. The Reconnex product captured and analyzed content transmitted over the internet using HTTP and other protocols, including storage protocols. Design of the Reconnex system was based on a modified and customized version of the Linux kernel which included specifically designed driver source code and interfacing between the application and kernel domains. Reconnex technology was employed in the audit and compliance verticals based on its ability to quickly analyze and classify the content of transmissions and interact with protocols and communications crossing corporate and departmental boundaries.

12. Between August 2007 and September 2008, I founded Strangways, a company focused on addressing messaging security for webmail and internet messaging protocols. Products and technology developed by Strangways were sold to SendMail Inc., Iron Port (Cisco), and other companies. The Strangways product utilized a Squid caching server and ICAP content rewrite server to translate HTTP to SMTP.

13. Between September 2008 and April 2014, I worked as Chief Architect for Gridiron Systems and, through acquisition, Senior Director of Technology for Violin Memory. I held the Senior Director position with Violin Memory both before and after the company's IPO. My responsibilities included architecture and

design of flash media controllers, adaptive caching design, and machine learning for application storage behavior analytics.

14. The Gridiron product was a storage accelerator used in many analytic environments such as big data architectures with constant ingest and processing of financial, consumer and IoT data. In my role as Chief Architect, I often worked with customers to implement and accelerate their analytics environments and spoke on related topics at trade shows and industry user groups. During product development, I worked with the data science team at Shopzilla to improve database and storage performance for real-time shopping recommendation and user tracking for advertising.

15. My work at Webstacks and Reconnex relate directly to packet processing including concepts such as priority and rules based processing relevant to the current proceeding. Approximately two dozen of my issued patents relate directly to these concepts.

16. Additional information regarding my employment history and prior expert testimony is included in Exhibit 2002. As reflected in Exhibit 2002, I have consulted on several matters involving intellectual property disputes including instances of alleged patent infringement requiring analysis of source code, reviewing and analyzing patents, writing expert reports, and testifying in court. I have testified at deposition and at trial in a number of these cases.

III. Legal Principles

A. Claim Construction

17. I understand that the first step in performing a validity analysis of the patent claims is to interpret the meaning and scope of the claims by construing the terms and phrases found in those claims. I understand that the appropriate construction of a claim term is its ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the entire patent and the prosecution history.

18. I understand that standard for claim construction in an *inter partes* review is the same standard as is applied in district court proceedings.

19. I understand that a determination of the meaning and scope of the claims is a matter of law. I have been informed that to determine the meaning of the claims, one should consider the intrinsic evidence, which includes the patent's claims, written description, and prosecution history.

B. Burden of Proof

20. I understand that in an *inter partes* review, the petitioner has the burden of proving unpatentability by a preponderance of the evidence.

C. Anticipation

21. I have been instructed by counsel and understand that a reference is anticipated if a single prior art reference discloses each and every claim element,

either explicitly or inherently, as arranged in the same way as in the claim. I understand that where even one claim element is not disclosed in a reference, a contention of anticipation fails.

22. I further understand that when a reference fails to explicitly disclose a claim element, that reference inherently discloses that element only if the reference must *necessarily* (or “inherently”) include the undisclosed claim element.

D. Obviousness

23. I have been instructed by counsel and understand that a combination of prior art references may render a claim obvious if, at the time of the invention, a person of ordinary skill in the art would have selected and combined those prior-art elements in the normal course of research and development to yield the claimed invention.

24. I understand that in an obviousness analysis, one should consider the *Graham* factors, including: the scope and content of the prior art; the differences between the claimed inventions and the prior art; and the level of ordinary skill in the art. I further understand the obviousness analysis is to be performed on a claim-by-claim basis. I understand that a person of ordinary skill in the art is a person of ordinary creativity, not an automaton.

25. I have been instructed by counsel and understand that obviousness requires more than a mere showing that the prior art includes separate references

covering each separate limitation in a claim under examination. I understand obviousness requires the additional showing that a person of ordinary skill at the time of the invention would have been motivated to combine those references in a manner that would include all limitations of the challenged claim, and, in making that combination, a person of ordinary skill in the art would have had a reasonable expectation of success.

26. I also understand that an obviousness analysis must be conducted with awareness of the distortion caused by hindsight bias and with caution of arguments reliant upon *ex post* reasoning. For instance, I understand that when considering obviousness, I should put myself in the position of a person of ordinary skill in the field at the time of the invention, rather than considering new information that is known today, but was not known before the priority date of the challenged patent.

IV. Person of Ordinary Skill in the Art

27. Dr. Wolfe proposes that a POSITA in the field of the '541 patent “would have had a minimum of a bachelor’s degree in computer science, computer engineering, or a similar field, and approximately two years of industry or academic experience in a field related to computer software development and/or computer networking. Work experience can substitute for formal education and additional formal education can substitute for work experience.” Ex. 1015, ¶51. For purposes of this my analysis, I do not disagree with this proposed level of skill

in the art. I meet at least these requirements of a POSITA, and have met those requirements since before the priority date of the '541 patent.

V. Claim Construction

28. I understand that Dr. Wolfe proposes no claim constructions in his declaration (Ex. 1015, ¶47), and for purposes of this declaration I also propose no constructions. Instead, I apply the plain and ordinary meaning of claim terms to a POSITA in light of the intrinsic record.

VI. Dr. Wolfe's challenge to claim 79 is unsupported.

29. Claim 79 requires that the application of the policy recited by claim 1 comprises “assist[ing] in intercepting a stack application programming interface (API) or application messaging layer request.” Dr. Wolfe's theory in this regard is that as part of policy application, Rao's filter 322 would “need to send a call (i.e., request) to the agent 326 through the API requesting that the agent 326 send the priorities,” which would constitute the claimed request or message. *See* Ex. 1015, ¶¶252-257 (alleging that “the request [from the filter to the agent] would constitute a ‘stack application programming interface (API) level’ request”).

30. Dr. Wolfe doesn't provide any obviousness theory with respect to claim 79—for example, he alleges in paragraph 252 of his declaration that “Rao teaches this limitation,” and also doesn't provide any allegation that a POSITA

would have been motivated to modify Rao in any way to arrive at the dependent limitation of claim 79. *See generally* Ex. 1015, ¶¶ 252-257.

31. Despite asserting that “Rao teaches this limitation,” Dr. Wolfe does not point to any express teaching in Rao that discloses the claim 79 limitation. For example, he does not identify any quotation from Rao that discusses filter 322 sending a call or request to agent 326 to retrieve priorities from agent 326 (which alleged call/request he relies on as being the claimed “stack application programming interface (API) level” request—*see* Ex. 1015 ¶256). Instead, I understand Dr. Wolfe to argue that Rao’s teaching of agent 326 communicating priorities to filter 322 **requires** filter 322 to first request those priorities. *See* Ex. 1015, ¶256 (alleging that “for determined priorities to be communicated from the agent 326 to the filter 322 so that they can be applied, the filter 322 would **need to** send a call (i.e., request) to the agent 326 through the API requesting that the agent 326 send the priorities”) (internal citation omitted).

32. As explained in the following sections, I disagree with Dr. Wolfe’s assertion in this regard, and it is also my opinion that Dr. Wolfe does not set forth any evidence that would even arguably show that Rao’s filter would need to **request** priorities from Rao’s agent before the agent could transmit those priorities to the filter.

A. A POSITA would have understood that Rao's filter need not send a request to agent 326 in order to retrieve the priorities.

33. As I noted above, I disagree with Dr. Wolfe's contention "for determined priorities to be communicated from the agent 326 to the filter 322 so that they can be applied, the filter 322 would *need to* send a call (i.e., request) to the agent 326 through the API requesting that the agent 326 send the priorities." Ex. 1015 ¶256 (internal citation omitted).

34. To the contrary, a POSITA would have understood that Rao's filter 322 would *not* need to (or even want to) issue a request to Rao's agent 326 before agent 326 communicated the priorities to the filter 322. Instead, a far more sensible, straightforward, and beneficial implementation would involve Rao's agent automatically communicating priorities to the filter *without* the filter having first requested those priorities.

35. For instance, as one of many potential implementations, a POSITA would have understood that Rao's priority transmission could be initiated by agent 326 every time a change to the assigned priorities of particular applications' packets occurs; on those occasions, agent 326 could transmit an updated set of those priorities to filter 322. A POSITA would understand that after receiving these priorities, filter 322 could update routing table 538 with the updated priority information, so that it could be efficiently accessed by the filter on an as-needed

basis. *See* Ex. 1004, ¶190 (teaching that “filter 322 [may not be] aware of the application 338a-338n by name but can associate priorities to network packets of applications 338a-338n via the routing table 538,” and accordingly, the agent can “indicate[] priorities to the filter 322 by routing information instead of application name” or “provide[] to the filter 322 a mapping between applications... to the routing information in a routing table 538”).

36. In my view, a POSITA would have recognized that configuring agent 326 to *automatically* send priority updates to filter 322 (without filter 322 needing to request those updates) would be preferable to the Petition’s alleged “require[d]” configuration, because agent 326 (in contrast to filter 322) would know when the priorities have been changed and thus know when updates to those priorities need to be transmitted to the filter. In contrast, filter 322 would simply be guessing when priorities had been updated. Having Rao’s agent 326 send updated priorities as they are available, rather than waiting for Rao’s filter 322 to request those priorities, would avoid inefficient requests from the filter (e.g., requests for priority updates when no priority updates are available).

37. I don’t believe a POSITA would have even considered implementing Rao’s system such that filter 322 would request updates from agent 326. But to the extent a POSITA were to consider such an implementation a POSITA would have recognized that such an approach would lead to significant drawbacks. For

example, if filter 322 were to request updates periodically (e.g., every 30 seconds), it would necessitate wasted processing cycles to request the updates from agent 326 and receive the agent's response, when the majority of responses would likely involve no update whatsoever. Such an implementation would further allow for delays between (1) the time when the priorities are updated at the agent and (2) the time when the updated priorities are actually communicated to filter 322.

38. I suppose that another hypothetical way to implement a system in which the filter requested priority updates from the agent might involve the filter requesting those updates on an "as-needed" basis, e.g., as individual packets are received by the filter. However, this would be *extraordinarily* costly from a processing power standpoint, because API requests and responses would need to be transmitted at very high frequency with no associated benefit. And if the filter 322 actually waited for the agent 326's priority response on a packet-by-packet basis, the application of priorities would operate exceptionally slowly, as each packet decision would be delayed while the filter and agent communicated amongst themselves. As of Rao's filing date (2005), systems such as Rao's devices would have been expected to process line-rate packet transmissions (for example, handling a gigabit per second of packets) and it would not be a realistic implementation for the agent to have to provide packet priorities every time a new packet is received (the filter's job would be to apply established priorities to

incoming packets based on features of those packets), much less for the filter to have to **request** packet priorities on every incoming packet, which would require a round-trip communication for each incoming packet. In my opinion, this would not be a realistic implementation at all because it would massively delay Rao's real-time packet filtering system, and a POSITA would certainly never design something like that (much less believe it to be required of Rao's teachings).

39. A POSITA would have known that the drawbacks resulting from implementing Rao's teachings as involving Rao's filter requesting priority updates from Rao's agent 326 would not have been present in a system where Rao's agent 326 automatically communicate the priorities to filter 322 as those priorities are updated. In such an implementation, there is no need for Rao's filter to request the updates, because they are automatically provided to Rao's filter as they are updated at Rao's agent. Furthermore, in such an implementation, there would be no wasted processing cycles or API communications associated with the packet priorities; agent 326 would **only** communicate the priorities to filter 322 when an update would actually provide helpful information to filter 322 (i.e., when the agent has new priorities that would cause the priorities previously received by the filter to be outdated).

40. Accordingly, because there are other (and clearly better) ways to implement Rao's teachings beyond having filter 322 request priorities from agent

326, Dr. Wolfe is incorrect in his assertion that “the filter 322 would need to send a call (i.e., request) to the agent 326 through the API requesting that the agent 326 send the priorities.” Ex. 1015 ¶256.

B. Dr. Wolfe provides no evidence to the contrary.

41. As explained above, it is my opinion that a POSITA would have recognized that Rao’s filter would not need to request priorities from Rao’s agent in order for the agent to communicate those priorities to the filter. I have reviewed Dr. Wolfe’s declaration on this point, and do not see any evidence or discussion that even arguably would result in a different conclusion.

42. For instance, Dr. Wolfe does not point to any teaching in Rao that suggests that Rao “the filter 322 would need to send a call... to the agent 326 through the API requesting that the agent 326 send the priorities.” *See* Ex. 1015, ¶256. The only citation to Rao in support of this assertion is a reference to Rao at ¶¶189-193, but neither that portion of Rao (nor any other portion of Rao) discusses filter 322 sending a request to Rao’s agent 326.

43. Instead, the cited paragraphs of Rao address, for example, how “agent 326 uses the policies 520 to apply a priority to network packets of applications” (Ex. 1004, ¶189), how Rao’s “agent 326 indicates the priority to the filter 322 for management of network packet queues 540a-540n to apply the indicated priorities... by any suitable means” (*id.*, ¶190), and the steps that filter 322 may

take “[b]ased on the indicated priorities for applications.” *Id.*, ¶191; *see also id.* ¶¶192-193. None of these paragraphs (or any other portion of Rao) say anything about filter 322 **requesting** the priorities from agent 326, which as explained above would not be something a POSITA would view as a reasonable implementation of Rao.

44. Dr. Wolfe further states that a “POSITA would further recognize that, in order to perform the callback functions that return the priorities to the filter 322, the agent 326 would need to at least in intercepting the filter 322’s stack API level request” (Ex. 1015, ¶256), and asserts that applications such as those used in Rao would “ordinarily include[] callback functions.” Ex. 1015, ¶257 (citing Wyld, Ex. 1005 7:36-40). However, Dr. Wolfe does not provide any reason why, even assuming Rao’s applications include “callback functions,”¹ that these “callback functions” would even allow for providing packet priorities (much less why any such alleged callback functions would be **necessary** to Rao’s communication of priorities from agent to filter).

¹ Dr. Wolfe acknowledges that these applications of Rao do **not** need to include callback functions, because he alleges only that those applications would “**ordinarily** include[] callback functions.” Ex. 1015, ¶257. I agree that there is no need for Rao’s applications to include callback functions, and especially no need for them to include callback functions that would be used to communicate packet priorities.

45. In my opinion, a POSITA would have viewed callback functions as inappropriate for providing priority updates to Rao's filter, because as I noted in the previous section, requiring the filter to request updates from the agent would range from inefficient to unworkable depending on the implementation. Specifically, as noted above, Rao's filter would frequently be requesting priority updates in many situations when there would be no priorities to update, and depending on the frequency of the update requests in such an implementation, Rao's filter would either be applying outdated priorities in many instances (in the case of periodic requests for updates), or would be requesting updates so frequently as to significantly impact system operation (in the case of requesting updated priorities on every incoming packet). A POSITA would have actively avoided such an implementation, and certainly wouldn't have viewed such an implementation as inherent in Rao's teachings.

46. My opinion that a POSITA would not have viewed callback functions as appropriate for providing updated packet priorities is supported by Dr. Wolfe's failure to identify any prior art that implemented such a solution. For instance, the only evidence that Dr. Wolfe cites for callback functions existing in the prior art is a function in the "Wyld" reference, wherein "a software component may request information, such as 'usage statistics on a given network connection.'" Ex. 1015 ¶257 (quoting Wyld, Ex. 1005 7:36-40). This callback function in Wyld is not used

to communicate priority information; rather, it is used to provide “usage statistics.” A POSITA would understand that requesting “usage statistics” is not the same as requesting packet priorities, and that the use of callback functions for the former does not suggest that call back functions should be used for the latter. Wyld’s discussion of a callback function for requesting usage statistics is certainly not evidence that provides any support for Dr. Wolfe’s assertion that Rao’s “filter 322 would *need* to send a call (i.e., request) to the agent 326 through the API requesting that the agent 326 send the priorities.” See Ex. 1015, ¶256.

47. None of Dr. Wolfe’s other opinions addressing claim 79 provide any reason why filter 322 would supposedly need to issue a request to agent 326. He asserts, for example, that “agent 326 may communicate the priority of network packets to the filter 322... via an application programming interface (API)” (Ex. 1015 ¶255, quoting Ex. 1004 ¶190), but this is irrelevant to whether the *filter* would need to send a request to the *agent* before the agent communicates the priorities. A POSITA would have interpreted this disclosure as suggesting that the agent would send the priorities to the filter *without* previously receiving a request from the filter, because that is the logical way to design a system such as Rao’s.

VII. Dr. Wolfe’s claim 83 challenge fails for at least the same reasons as his claim 79 challenge.

48. Dr. Wolfe simply incorporates his claim 79 analysis into his claim 83 invalidity assertion. See Ex. 1015 ¶270 (alleging: “Rao teaches this limitation for the reasons presented in connection with claim 79.”). As explained above, Dr. Wolfe’s “reasons presented in connection with claim 79” do not support his assertion that Rao teaches the limitation recited by claim 79, and it is my opinion that Dr. Wolfe’s claim 83 theory fails for at least the same reasons.