

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD.,
Petitioner,

v.

HEADWATER RESEARCH LLC,
Patent Owner.

Case IPR2024-01407
Patent 9,179,359 B2

**DECLARATION OF DR. MICHAEL C. BROGIOLI IN SUPPORT OF
PATENT OWNER'S RESPONSE**

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.....	7
D.	Obviousness	7
IV.	PERSON OF ORDINARY SKILL IN THE ART	9
V.	CLAIM CONSTRUCTION	9
VI.	OVERVIEW OF THE '359 PATENT	10
VII.	OPINIONS REGARDING CLAIM 1 STANDING ALONE.....	12
A.	Dr. Butler fails to show obviousness of denying or blocking piggy back network connection requests.	12
B.	Dr. Butler fails to show that Shell-Cole would indicate unavailability of the alleged Internet data service to an application, while the same Internet data service is available to another application.	20
VIII.	OPINIONS REGARDING CLAIM 26.....	22

I, Michael C. Brogioli, PhD, 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 Headwater Research LLC (“Headwater”) for the above-captioned *inter partes* review (“IPR”). I understand that the petition for *inter partes* review involves U.S. Patent No. 9,179,359 (“the ’359 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. I am being compensated at a rate of \$750 per hour for my time in this matter. My compensation does not depend on the outcome of this proceeding, and I have no financial interest in its outcome.

5. In preparing this Declaration, I have reviewed and considered the ’359 patent, the ’359 patent’s prosecution history, the Petition, Dr. Butler’s declaration submitted in this proceeding, Dr. Butler’s deposition transcript, 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 2013. That exhibit also includes a list of my publications.

7. I am currently an Adjunct Professor of Electrical and Computer Engineering at Rice University in Houston, TX, teaching Ph.D. candidate level courses in wireless telecommunications, embedded computing software, embedded computing hardware, and software/hardware design. I have held this position since July 2009. I am also currently the Managing Director of Polymathic Consulting in Austin, TX, a company that advises a variety of clients on engineering, research and development, intellectual property, and other technical leadership matters, and I have held this position since October 2011.

8. I am a named inventor on multiple issued and pending patent applications. The subject matter of these applications includes, but is not limited to, embedded processing, edge computing, and various aspects of distributed systems including computation, interconnectivity and hardware/software design and implementation.

9. From November 2009 to October 2011, I was Chief Architect, Senior Member Technical Staff at Freescale Semiconductor in Austin, TX (formerly Motorola), responsible for management of technology, engineering roadmaps,

design lead on compiler infrastructure and optimizations for mobile and embedded computing solutions, including the types of security and encryption technologies involved in this matter. From 2007 to 2009, I was a Senior Engineer working with high performance compiler design and next generation baseband compiler design and next generation baseband microprocessor and microcontroller architecture at Freescale Semiconductor in Austin, Texas, including regarding communications networks and how mobile device architecture and design interacts with network elements.

10. From June 2006 to August 2007, I worked as the Technical Co-Founder of Method Seven LLC, in Boston, MA, working with high performance software and hardware systems architecture. I am currently the co-founder, co-inventor, and Chief Technology Officer of Network Native, an Internet of Things hardware and software company. I have also served on multiple advisory boards for technology companies over the past 10 years.

11. While at Rice University, I developed various computer architecture designs. For example, from 2002 to 2004, I developed Spinach, a simulator design toolset for modeling programmable network interface architectures, which models system components common to all programmable computing environments as well as components relating to embedded computing, wireless and mobile devices. From 2004 to 2009, I developed Spinach DSP-FPGA, a modular and composable

simulator design infrastructure for programmable and reconfigurable embedded SOC architectures specifically targeting mobile, embedded and portable computing devices. From 2005 to 2009, I developed and published a retargetable compiler infrastructure for software and systems related to mobile and embedded computing technologies. Many of these tools have been used at U.S. universities in the area of electrical and computer engineering research.

12. I am recognized as an expert in the field of communications systems, including computer architecture, hardware, and software as they relate to the subject matter at hand. I am a member of the Institute of Electrical and Electronics Engineers (IEEE). I am currently the Program Chair of Design Automation Conference in the areas of Embedded and Wireless Solutions. I have been a Program Committee member for the IEEE and ACM Design Automation Conference from 2011 to the present, as well as other IEEE and ACM conferences. I have published, co-published and contributed to numerous works on wireless and mobile computing, digital networking, Internet of Things technology, software and hardware optimizations, and digital signal processing and related architectures among other topics.

13. I have previously served as an engineering consultant and testifying witness on matters related to, and including Internet and communications technologies such as those at hand in this matter. This experience is encompassed

in my work in academia and as a faculty member, my time in industry as a system designer and architect, as well as my work as an engineering consultant and expert witness over the past 15 years.

14. Over the past 20+ years, I have authored numerous peer reviewed publications in the area of embedded and high performance computing, interconnect technologies, computer networking at the server and mobile handset level, and optimization at the hardware and software design level. I have also been a committee member, and chair of Design Automation Conference in areas of technology related to the '359 Patent, often reviewing and vetting submissions from major industry players often related to technologies at issue in this matter.

15. During my tenure at Freescale Semiconductor, I was in charge of system developer tools for baseband processor design, both in the hardware and software spaces. This work included hardware and software design issues of next generation wireless networking equipment, including system level programming models, interconnect technology, heterogeneous computing, various hardware acceleration technologies and related components.

16. My current curriculum vitae, provided in Exhibit 2013, contains more information on my background and experience, as well as the cases in which I have served as an expert witness the past four years.

17. No part of my compensation is contingent upon the outcome of this litigation. I have no other interests in this litigation or with any of the parties.

III. LEGAL PRINCIPLES

A. Claim construction

18. 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.

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

20. 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

21. I understand that in an *inter partes* review, the petitioner has the burden of proving unpatentability by a preponderance of the evidence. I further understand that for an *inter partes* review to be instituted, the petitioner must show

a reasonable likelihood that the petitioner would prevail with respect to at least one of the claims challenged in the petition.

C. Anticipation

22. 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.

23. 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* include the undisclosed claim element.

D. Obviousness

24. 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.

25. I understand that in an obviousness analysis, one should consider the *Graham* factors: the scope and content of the prior art; the differences between the claimed inventions and the prior art; the level of ordinary skill in the art; and

certain secondary considerations, identified below. 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.

26. 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*.

27. 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.

28. In particular, I understand that it is improper to use the challenged patent's disclosure or invention as a roadmap to find its prior-art components,

because such an approach discounts the value of combining various existing features or principles in a new way to achieve a new result.

IV. PERSON OF ORDINARY SKILL IN THE ART

29. Dr. Butler states that a POSITA would have had “(1) at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or a related field, and (2) at least two years of industry experience in services and application implementation in communication networks. Additional advanced education could compensate for a deficiency in work experience, and *vice versa*.” Ex. 1003, ¶15.

30. For purposes of this declaration, I do not dispute Dr. Butler’s proposed level of skill in the art.

31. I further note that I am a person of ordinary skill in the art, and was such a person as of the priority date of the ’359 patent.

V. CLAIM CONSTRUCTION

32. Dr. Butler does not propose that any terms be construed. For each claim limitation at issue, I apply what I believe would have been the plain and ordinary meaning to a POSITA in light of the specification and file history of the ’359 patent.

VI. OVERVIEW OF THE '359 PATENT

33. The challenged claims of the '359 patent (claims 1 and 26)¹ are generally directed to an end-user device that indicates unavailability of WWAN access to one application while WWAN access is available to a different application, wherein the “different application” has been determined by the system to be an “active application based on a user interaction with the different application.” Claims 1 and 26 are reproduced below.

1. A wireless end-user device, comprising:

- [1.1] a wireless wide area network (WWAN) modem to receive and transmit Internet data between the device and at least one WWAN when configured for and connected to the at least one WWAN;
- [1.2] a wireless local area network (WLAN) modem to receive and transmit Internet data between the device and at least one WLAN when configured for and connected to the at least one WLAN; and
- [1.3] one or more processors configured to apply a stored WWAN differential traffic control policy to Internet data service provided using the WWAN modem and the at least one WWAN, and
- [1.4] indicate to a particular application, via an application program interface (API), one or more network access conditions based on the applied WWAN differential traffic control policy,

¹ I understand all claims except claim 26 have been disclaimed, but that claim 1 is still relevant because claim 26 depends from it.

[1.5] wherein the one or more network access conditions includes a network access condition that indicates the unavailability to the particular application of an Internet data service that is currently available via the WWAN modem to a different application, but that is indicated as unavailable to the particular application.

26. The wireless end-user device of claim 1, *wherein the one or more processors are further configured to determine that the different application is an active application **based on a user interaction with the different application.***

Ex. 1001, claims 1, 26.

34. As shown above, Claim 26 states that the system includes “one or more processors” configured to “determine that the different application” (i.e., the application for which the Internet data service *is* currently available to as recited in Claim 1) “is an active application based on a user interaction with the different application.” Claim 26’s requirements thus further expand on the requirements of claim 1, because the “different application” is the application to which “Internet data service... is currently available via the WWAN modem,” while the system “indicates the unavailability” of the same Internet data service to “a particular application” (distinct from the claimed “different application” determined to be “an active application”). A POSITA would thus understand that, because Claim 26 recites that it is specifically the “different application” that is determined to be an “active application” in the recited manner, that Claim 26’s active application determination impacts how claim 1’s “differential traffic control policy” is applied,

including affecting the application to which “Internet data service... is currently available.”

VII. OPINIONS REGARDING CLAIM 1 STANDING ALONE

35. Dr. Butler alleges that limitation [1.5] is obvious in view of the combined Shell-Cole system “denying/blocking [a] ‘piggy back’ network connection request.” *See* Ex. 1003 ¶¶103-105. As explained below, I disagree that this would have been an obvious implementation, and I also disagree that the combination would satisfy claim 1 under Dr. Butler’s theory.

A. Dr. Butler fails to show obviousness of denying or blocking piggy back network connection requests.

36. Dr. Butler analysis regarding how denying or blocking a “piggy back” request satisfies the relevant limitations is reproduced below:

104. Shell further describes examples in which an application may “wish[] to ‘piggy back’ on other connection requests[,]” i.e., “an application may request to only access resources in a destination network when another application is already accessing resources.” SAMSUNG-1005, [0070]. A POSITA would have recognized and/or found obvious that a “piggy back” connection request may be denied in some instances, e.g., due to limited network bandwidth. *See id.*, [0092] (“application modules may be notified that...a connection to a resource is blocked for some reason). In these instances, a first application (*the particular application*) sends a “piggy back” network connection request based on an established connection of a second application (*different application*). In denying/blocking the “piggy back” network connection request, the connection manager modules would provide notification data to the first application indicating decision information regarding the “piggy back” network connection request (*network access condition*). Further, the decision information would indicate that the “piggy back” network connection request is denied/blocked

(Internet data service being indicated as unavailable to the particular application), but that network access by the first application is available (Internet data service [] is currently available via the WWAN modem to a different application). Id.

Ex. 1003, ¶104 (emphases in original). I note that Dr. Butler presents no theory regarding how the requirements of limitation [1.5] are satisfied *other* than by denying or blocking a “piggy back” connection request; for example, he does not explain how claim 1’s requirements relating to the “different application” (including that the Internet data service be available to that “different application”) are met, other than in the context of a piggy back request. *See generally* Ex. 1003, ¶¶103-105.

37. For context, Shell teaches that its “piggy back” request occurs when an “application... notif[ies] connection planner 211 it wishes to ‘piggy back’ on other connection requests”:

In other embodiments, an application may notify connection plan[n]er 211 it wishes to “piggy back” on other connection requests. That is, an application may request to only access resources in a destination network when another application is already accessing resources. For example, sync engine 202 may wish to synchronize an electronic mail account only when a connection to the Internet has been established for another application. Applications that wish to piggy back, may call APIs associated with connection manager modules 210 and be given an open handle. The open handle facilitates notification back to the application that a connection on which the application may want to piggy back has been established.

Ex. 1005, [0070] (emphasis added). A POSITA would thus understand that a “piggy back” connection request is simply a request to open a handle, which will notify the application at a later time, whenever a connection suitable to piggyback upon—i.e., “a connection on which the application may want to piggy back”—has been established. I note that this understanding is also consistent with Dr. Butler’s testimony at deposition, in which “a piggyback request may call APIs to open the handle” such that the application can be informed in the future when a suitable connection is established. Ex. 2014 at 11:14-12:6 (“[I]n the context of Shell’s Paragraph 70, is the API call that is made as part of a piggyback connection request the same API call that results in the creation of an open handle? A.... [C]ertainly, there is a way in which a piggyback request may call APIs to open the handle.”).

38. In other words, an application’s piggyback connection request is an application’s request (sent to connection planner 211 via an API) to open a handle, such that “[t]he open handle facilitates notification back to the application that a connection on which the application may want to piggy back has been established.” Ex. 1005, [0070].

39. A POSITA would recognize that these piggyback connection requests have several advantages. As an initial matter, by connecting to the network only when another application is accessing the network, significant power savings can be achieved by reducing the number of times the network modem switches states

(e.g., active vs inactive, or transmitting vs not transmitting). For example, a modem may have to only switch from inactive to active a single time to process two transmission requests if piggy backing is used, as opposed to needing to switch two times if piggy backing is not used. Additionally, because it receives information about when a suitable connection is available for the application to piggyback upon, the application can know to initiate a network connection only when conditions are suitable for doing so. This prevents scenarios where the application might attempt to initiate a connection under less-than-ideal circumstances, such as when no other connection exists (which may be inefficient by causing additional power state changes in the modem), or when no connection can be established (such as when there is insufficient bandwidth for the connection to be opened).

40. In light of these advantages, I do not believe a POSITA would have found it obvious to ***block or deny*** an application's piggyback connection request (i.e., a request to open a handle so that the application can be informed when a connection meeting the application's needs ultimately becomes available).

41. Dr. Butler supposes that a piggyback connection request may be blocked or denied "due to limited network bandwidth." Ex. 1003, ¶104. But "limited network bandwidth" is a state in which the *current* network bandwidth is not suitable. And a *current* lack of bandwidth is no reason to deny an application's

request to open an API handle so that it can be informed *in the future* when network bandwidth *is* suitable for the application to access the network in the manner requested. The point of a piggyback connection request is not to *immediately* establish a connection, but rather to request an open handle so that the application can be informed (potentially in the future) whenever conditions are ideal for establishing a piggyback connection.

42. In fact, I believe a POSITA would be strongly motivated to *not* deny or block piggyback connection requests, because a POSITA would understand that, by connecting to the network only when another application is accessing the network, significant power savings can be achieved by reducing the number of times the network modem switches states (e.g., active vs inactive, or transmitting vs not transmitting). A POSITA would understand that if a piggyback connection request (which opens a handle informing the application when ideal conditions for a piggyback connection arise) is denied, the application would be prevented from receiving an open handle such that it can learn when ideal connection conditions exist for establishing a piggyback connection. Without such an open handle, the application would attempt network access *regardless* of preexisting connections, which would unnecessarily increase power consumption of the device.

43. A POSITA would have also understood that such indiscriminate network access attempts would frequently initiate connections to the network even

when no other connections existed, thus frequently switching a modem from an inactive state to an active state for the purpose of the new connection and causing significantly more battery drain than the application “piggybacking” on a preexisting connection. Accordingly, a POSITA would avoid blocking or denying piggyback connection requests, as doing so would lead to significantly more power inefficiency than if the piggyback connection requests were allowed.

44. I further note, that to the extent Dr. Butler’s position is that connection requests generated as a **result of** the application receiving a notification from the open handle that suitable conditions exist for the application’s connection, I do not believe it would be obvious for **those** requests to be blocked or denied either. Dr. Butler’s only support for the proposition that a “‘piggy back’ connection request may be denied in some instances” is a citation to paragraph [0092] of Shell (*see* Ex. 1003, ¶104), but paragraph [0092] does not disclose denying either a request to open a handle or a request for network access responsive to the requesting application receiving an open handle notification from Shell’s connection manager modules 210 that a suitable connection for piggy backing is available.

45. In my opinion, a POSITA would not have found it obvious to deny an application’s connection request that was responsive to an open handle notification that a suitable connection would be available to the application. Specifically, a POSITA would understand that, in the context of Shell’s piggy back teachings, that

the connection manager modules 210 would only notify the application that a connection is available *if and when* a connection would be allowed if requested. For example, there would be no reason for Shell's connection manager modules to inform an application via an open handle that it would be a good time for the application to initiate a connection, only for the connection manager modules to immediately thereafter deny a resulting connection request. This would lead to unnecessary and inefficient application behavior (e.g., waking an inactive application, causing that application to initiate a futile connection request, and have the connection manager modules 210 process unnecessary connection requests that could have been avoided if the appropriate open handle behavior had been implemented), and Dr. Butler does not allege that his proposed combination would operate in such a manner (much less provide a motivation to do so).

46. A motivation to avoid informing potentially piggybacking applications of a network connection upon which that the application **would not** be able to piggyback is supported by Shell itself. Specifically, Shell teaches that an "open handle facilitates notification back to the application that *a connection on which the application may want to piggy back has been established.*" Ex. 1005, [0070]. Of course, if a connection initiated by a different application is on a network where insufficient bandwidth exists to handle both the different application's connection and the potentially piggybacking application's

connection, a POSITA would not understand that to be a “connection on which the application may want to piggy back,” because no application would want to piggy back on a connection on which it would be unable to piggy back. As such, it would only be obvious for applications to be informed that a connection is possible via the open handle when a connection would actually be *allowed*. It certainly would not be obvious for a POSITA to implement a system that would *deny* or *block* any connection attempts initiated as a result of receiving a piggyback open handle notification, because receiving a notification via an open handle that a piggyback connection is possible would indicate that responsive connection attempts will be *allowed*.

47. As such, I do not believe it would be obvious, for example, for a POSITA to implement the combined Shell-Cole system such that code fragment discussed in Flanagan (Ex. 1011, not part of Dr. Butler’s proposed combination) would result in a denial of a connection pursuant to a notification by an open handle that a connection suitable to piggyback upon is available, while Internet access would remain available to the “different application.” If an application were attempting to piggy back on a pre-existing communication as a result of an open handle notification to that effect, the connection request would be allowed because the system would already have decided that the network is suitable for the requested piggy backing.

B. Dr. Butler fails to show that Shell-Cole would indicate unavailability of the alleged Internet data service to an application, while the same Internet data service is available to another application.

48. Claim 1 requires “one or more processors configured to... indicate to a particular application, via an [API], one or more network access conditions..., wherein the one or more network access conditions includes a network access condition that indicates the unavailability to the particular application of *an Internet data service that is currently available via the WWAN modem to a different application, but that is indicated as unavailable to the particular application.*” A POSITA would thus understand that the *same* “Internet data service that is currently available via the WWAN modem to a different application” must be also “indicated as unavailable to the particular application.”

49. Dr. Butler does not identify a consistent “Internet data service... that is indicated as unavailable to the particular application,” which same Internet data service is “currently available via the WWAN modem to a different application.” Instead, Dr. Butler asserts that a “**‘piggy back’ network connection request**” would be “denied/blocked [to the particular application]..., but that **network access** by the [different] application is available.” Ex. 1003, ¶104. In other words, Dr. Butler identifies two *different* Internet data services for its claim mapping: (1) a piggyback network connection request made by the “particular application,” and

(2) general “network access” by the “different application.” It is thus my opinion that Dr. Butler’s mapping satisfies the claim requirement of “indicat[ing] the unavailability to the particular application *of an Internet data service that is currently available via the WWAN modem to a different application*,” because Dr. Butler points to two *different* Internet data services as being (1) “currently available” to the different application and (2) indicated as unavailable to the particular application.

50. For instance, Dr. Butler does not allege, for example, that the combined Shell-Cole system indicates unavailability of *all* WWAN network access to the “particular application” that issued a piggyback connection request. Indeed, under Dr. Butler’s theory, a POSITA would understand that “the particular application” (e.g., a “Phone” application) might have certain network requests denied (e.g., requests that do not involve voice communications), whereas other network requests (e.g., those that *do* involve voice communications) would be granted because they are given higher priority. *See* Ex. 1005, [0057] (discussing that “voice requests may be assigned a higher priority than data requests,” separately from assigning priorities based on the identity or state of the application making the requests); Ex. 1003, ¶140 (discussing prioritization of voice-related connection requests over data requests). Dr. Butler does not explain why an indication that a *particular* connection request (e.g., a request to download updated

contact information) is denied or blocked would be an indication that *WWAN network access* more generally is unavailable to the particular application. Nor does Dr. Butler allege that the “different application” would have access to the same “‘piggy back’ network access request” made by the “particular application” whose piggyback request was denied or blocked.

51. As such, it is my opinion that Dr. Butler does not set forth any plausible theory how the *same* Internet data service is both (1) available to the different application, and (2) indicated as unavailable to the particular application according to his theory.

VIII. OPINIONS REGARDING CLAIM 26

52. As summarized above in Section VI, claim 1 recites:

1. A wireless end-user device, comprising:

- [1.1] a wireless wide area network (WWAN) modem to receive and transmit Internet data between the device and at least one WWAN when configured for and connected to the at least one WWAN;
- [1.2] a wireless local area network (WLAN) modem to receive and transmit Internet data between the device and at least one WLAN when configured for and connected to the at least one WLAN; and
- [1.3] one or more processors configured to apply a stored WWAN differential traffic control policy to Internet data service provided using the WWAN modem and the at least one WWAN, and

[1.4] indicate to a particular application, via an application program interface (API), one or more network access conditions based on the applied WWAN differential traffic control policy,

[1.5] wherein the one or more network access conditions includes a network access condition that indicates the unavailability to the particular application of an Internet data service that is currently available via the WWAN modem to a different application, but that is indicated as unavailable to the particular application.

Ex. 1001, Claim 1. In other words, claim 1 specifies that there is a “WWAN differential traffic control policy” that is applied “to Internet data service provided using the WWAN modem.” Claim 1 further specifies that the policy applies differently to two different applications. To a first application (the “particular application”), an Internet data service is indicated as unavailable, and to a second application (the “different application”) the same Internet data service is made “currently available.”

53. A POSITA would understand that Claim 26 further expands on the operation of claim 1, in specifying that the “different application” (i.e., the application to which Internet data service is made available via the applied policy) was determined to be an “active application” “based on a user interaction” with that application.

26. The wireless end-user device of claim 1, *wherein the one or more processors are further configured to determine that the different*

application is an active application based on a user interaction with the different application.

Ex. 1001, Claim 26. Taking claims 1 and 26 together, a “user interaction with the different application” is used as an input to make an active application determination, thereby influencing the policy such that “the different application” is provided access to the claimed Internet data service.

54. Dr. Butler contends that this requirement is satisfied because, “[i]n Shell, connection manager modules may assign priorities to network connection requests ‘based on the current state of applications associated with the connection requests.’” Ex. 1003, ¶140 (quoting Ex. 1005, [0057]). The quoted paragraph of Shell, paragraph [0057], teaches that an example of prioritizing connection requests “based on the current state of applications” includes prioritizing “a request from an application that is currently active”:

Priorities may also be assigned to connection requests based on the current state of applications associated with the connection requests. For example, *a request from an application that is **currently active** may be assigned a higher priority than a request from an application running in the background.*

Ex. 1005, [0057].

55. What Shell does not teach, however, is that a determination that the different application “is an active application” would be “*based on a **user interaction*** with the different application.” Dr. Butler thus relies on Shell’s

teaching that “voice requests may be assigned a higher priority than data requests,” and on that basis asserts in conclusory fashion that “[a]n application that provides a voice-related connection request... involves a determination that a user is likely to interact with the application.” Ex. 1003, ¶140 (quoting Ex. 1005, [0057]).

56. I disagree, however, that this disclosure teaches or renders obvious determining an “active application” based on “user interaction with [that] application” in the manner recited by claim 26. Dr. Butler’s declaration fails to mention that assigning “voice requests” a higher priority than “data requests” is described by Shell as an *alternative* prioritization algorithm to assigning priorities based on “the current state of applications” (an example of which is assigning priorities to applications that are “currently active”). Specifically, Shell teaches three different ways to prioritize connection requests (“based on applications associated with the connection requests,” “based on the current state of applications associated with the connection requests,” and “based on characteristics associated with the connection requests”):

Illustrated in FIG. 5A are connection requests 500 including new connection request 510 and active connection requests 520 and 530. Connection requests 510, 520, and 530 are prioritized, with connection 510 having the highest priority and connection 530 having the lowest priority. *Priorities may be assigned to connection requests based on applications associated with the connection requests.* For example, requests from sync engine 202 may be assigned a higher priority than requests from messaging application 206. *Priorities may also be assigned to connection requests based on the current state of*

applications associated with the connection requests. For example, a request from an application that is currently active may be assigned a higher priority than a request from an application running in the background. *Priorities may also be assigned to connection requests based on characteristics associated with the connection requests.* For example, voice requests may be assigned a higher priority than data requests. It should be understood that these are only examples of assigned priorities. It would be apparent to one skilled in the art, after having reviewed this description, that priorities may be assigned to connection requests using a variety of methods.

Ex. 1005, [0057]. A POSITA would understand that these three concepts (prioritizing connection requests (1) “based on applications associated with the connection requests,” (2) “based on the current state of applications associated with the connection requests,” and (3) “based on characteristics associated with the connection requests”) are *different* ways to prioritize connection requests. While a POSITA would understand that they can be used together (e.g., a voice request issued by an active application may have a higher priority than a voice request issued by an inactive application), nothing in Shell discloses or renders obvious that ““characteristics associated with the connection requests” (e.g., whether the requested connection is voice-related) would be used to determine “the current state of applications.”

57. Nothing in Dr. Butler’s analysis changes my opinion in this regard. For instance, he does not set forth any alleged *motivation* (or other obviousness rationale) to combine Shell’s different techniques such that the type of voice

connection would impact an active application determination. As such, I do not believe implementing Shell as Dr. Butler alleges—such that a specific type of connection request (such as a voice-related connection request) would be used to determine whether the requesting application “is an active application” as recited by claim 26—would have been obvious.

I declare under penalty of perjury 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.

June 27, 2025

A handwritten signature in dark ink, reading "Michael C. Brogioli", is written over a horizontal line.

Michael C. Brogioli, Ph.D.