

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 IPR2024-00003
Patent 9,198,117

**DECLARATION OF DR. MICHAEL C. BROGIOLI IN SUPPORT OF
PATENT OWNER'S PRELIMINARY 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.	ANALYSIS OF DR. TRAYNOR’S GROUND 1	9
VII.	ANALYSIS OF DR. TRAYNOR’S GROUND 2	19

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. 9,198,117 (“the ’117 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 ’117 Patent, the ’117 Patent’s prosecution history, the Petition, Dr. Traynor’s declaration submitted 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. 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 specific to embedded. 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 15+ years, I have authored numerous peer reviewed publications in the area of embedded and high performance computing, interconnect technologies, 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 '680 Patent, often reviewing and vetting submissions from major industry players in the design and architectural issues related to various 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 2002, 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. Traynor states that “a POSITA as of the filing date of the effective filing date of the ’241 Patent (January 28, 2009) “(1) at least a bachelor’s degree in computer science, electrical engineering, or a related field, and (2) 3-5 years of experience in services and application implementation in communication networks. Additional graduate education could substitute for professional experience, and vice versa.” Ex. 1003, ¶21. For purposes of this declaration, I do not dispute Dr. Traynor’s proposed level of skill in the art. I further note that I am a person of ordinary skill in the art under Dr. Traynor’s proposal, and was such a person as of January 28, 2009.

V. CLAIM CONSTRUCTION

30. Dr. Traynor does not propose a construction of any term at issue in this proceeding. 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 ’117 Patent.

VI. ANALYSIS OF DR. TRAYNOR’S GROUND 1

31. Limitation [1.3] of Claim 1 (according to the Petition and Dr. Traynor’s numbering convention for claim limitations) recites: “the network

message server configured to receive, from each of a plurality of network application servers, multiple requests to transmit application data, each such request indicating a corresponding one of the mobile end-user devices and one of a plurality of applications.” Ex. 1001, Claim 1. Claim language thus states that requests sent *from* application services and *received by* the message server must indicate both (1) “a corresponding one of the mobile end-user devices” and (2) “one of a plurality of applications.”

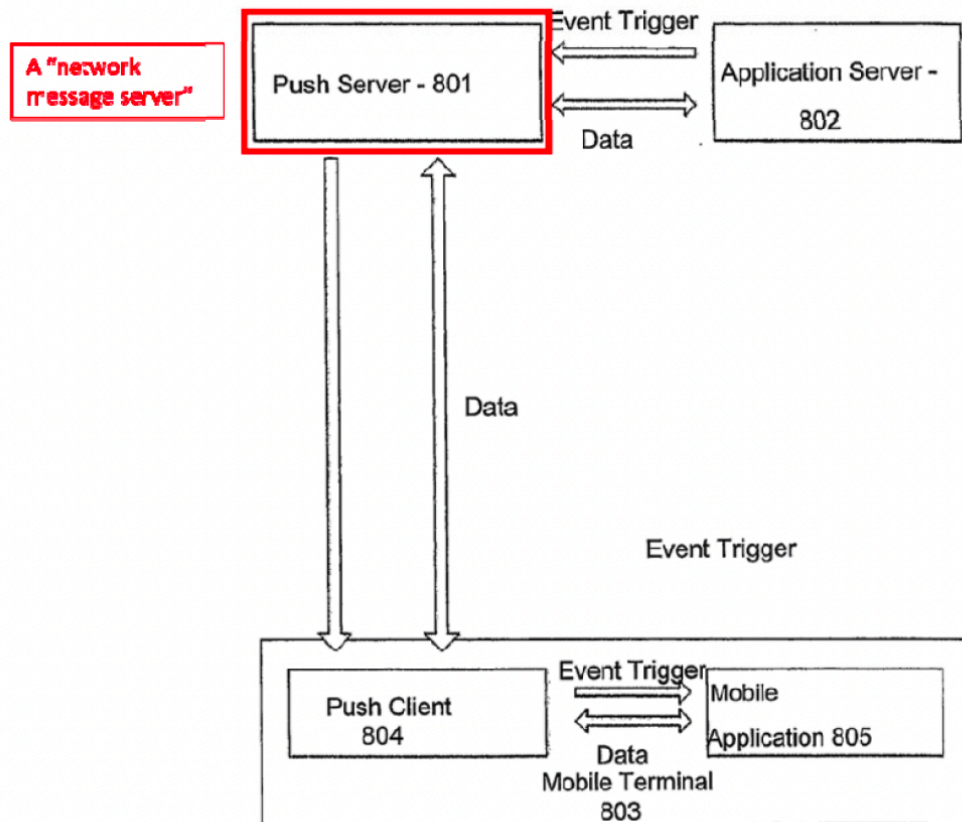
32. I understand that for Ground 1A, the Dr. Traynor relies on the Houghton reference alone as disclosing or rendering obvious limitation [1.3]’s requirement that each of multiple requests to transmit data received from each of a plurality of network application servers indicates both “a corresponding one of the mobile end-user devices” and “one of a plurality of applications.” *See* Ex. 1003 ¶¶46-47. While Dr. Traynor does discuss Kalibjian and Munson in discussing this limitation, he does so only with respect to the requirement that the claimed system include a plurality of application servers.

33. As I discuss in more detail in the following paragraphs, I disagree with Dr. Traynor’s opinion that Houghton discloses that the claimed requests sent by the application server and received by the message server indicate both “a corresponding one of the mobile end-user devices” and “one of a plurality of applications.” Additionally, because Dr. Traynor does not set forth any proposed

modification to the information those requests would contain, I do not understand Dr. Traynor to set forth any obviousness theory with respect to these requirements.

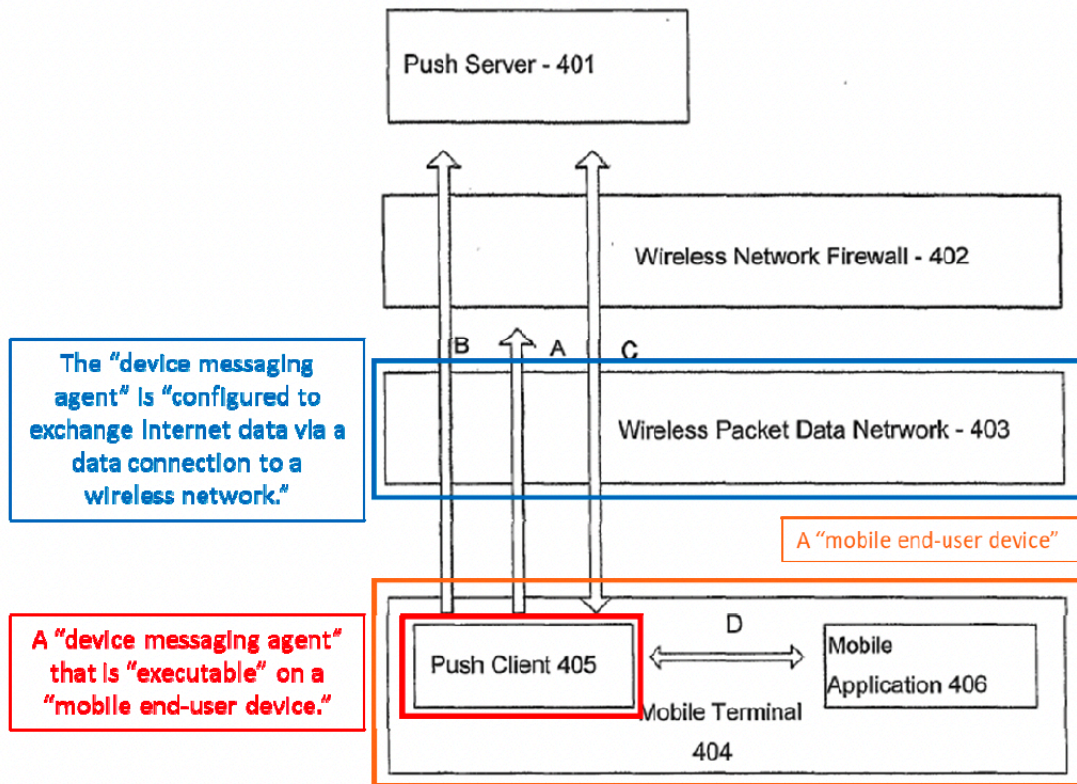
34. Before addressing my opinions regarding why I disagree with Dr. Traynor's opinion that Houghton discloses the requirements of limitation [1.3], it is first helpful to provide an overview of Dr. Traynor's mapping of the prior art to claim 1 more broadly.

35. Specifically, Dr. Traynor maps the "network message server" of claim 1 to the "push server" in Houghton (which Dr. Traynor acknowledges is indicated in Houghton by alternate designations 202, 301, 401, 501, 701, 801, and 901). *See* Ex. 1003, ¶41 & n.5. Dr. Traynor's identification of the "push server" as being the claimed "network message server" is reproduced below.



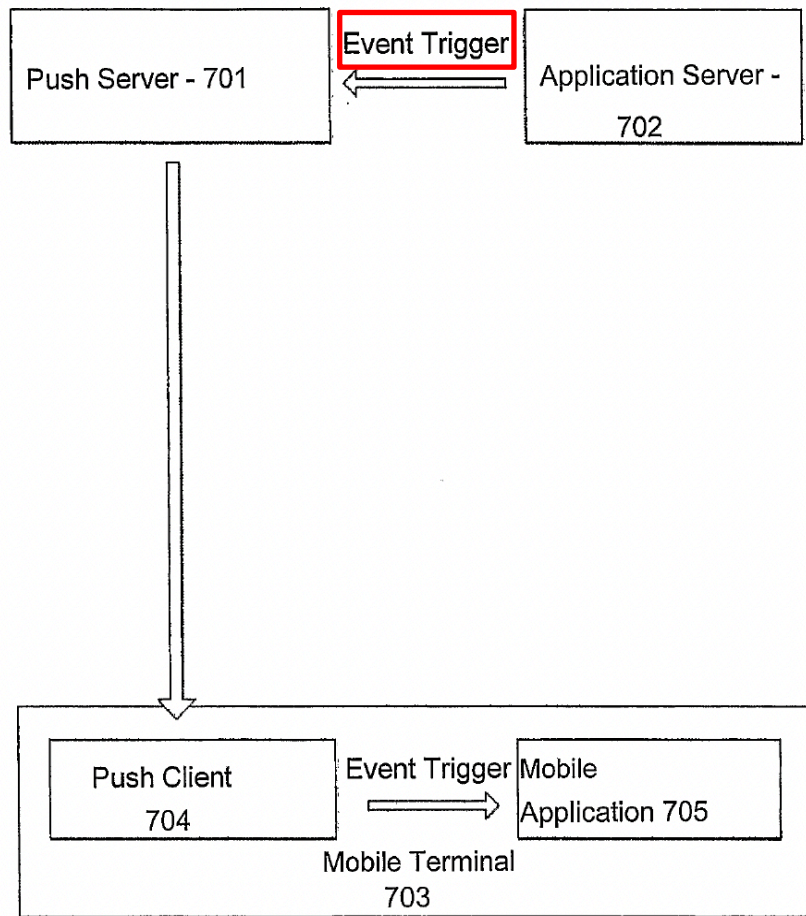
36. Ex. 1003 at ¶41 (Dr. Traynor’s annotation of Houghton’s FIG. 8).

37. Dr. Traynor additionally maps the “device messaging agent” of the claim as being Houghton’s Push client,” which residents in Houghton’s “mobile terminal” (which Dr. Traynor separately maps to the claimed “mobile end-user device”). *See* Ex. 1003 ¶38. Dr. Traynor’s identification of these elements is reproduced below.



38. Ex. 1003 ¶38 (Dr. Traynor’s annotation of Houghton’s FIG. 4).

39. Dr. Traynor further contends that a “trigger event as for example from an application server 702” is the claimed “request[] to transmit application data” received from an application server. Ex. 1003 ¶46. An example of this trigger event (which Houghton alternatively refers to as an “event trigger”) sent from the application server to the push server is provided in my annotation of Houghton’s FIG. 7 below:



40. Houghton discloses that the “trigger events” sent by the application server 702 and received by the push server 701 trigger the push server to push a command message to push client 704. Ex. 1005, 21:35-22:4 (“Figure 7 illustrates an example of a[] ‘COMMAND PUSH’ procedure where *the push server 701 is triggered by a trigger event* as for example from an application server 702 *to push an application command message to the push client 704* and thereby initiate a mobile terminal client trigger event in a mobile application from a plurality of such

applications, program modules and user interface features on the terminal 705.”¹; *see also* Ex. 1003 ¶48 (acknowledging that “the push server generates ‘command’ and ‘data push messages’ for transmission... to the push client in response to a ‘trigger event’ received from an ‘application server 702’”). I note that Houghton’s trigger event / event trigger sent from the application server to the push server is different than the trigger event / event trigger sent from the push client to the mobile application. *See* Ex. 1005, 21:25-29, 21:35-22:6 (separately discussing the two types of trigger events). Both of these event triggers are also distinct from the command message (or push message) the push server sends to the push client. For example, Houghton teaches that the “push message from the push server 401 may be **triggered** by any trigger event,” thus making clear that the trigger event is merely a trigger for the push message and is not the same as the push message. *See* Ex. 1005, 21:25-22:4. Likewise, Houghton teaches that the “application command message” that is transmitted as part of the push message “initiate[s] a mobile client trigger event in a mobile application 705,” again making clear that the command / push message **initiates** a trigger event sent from the push client to the mobile application and thus is distinct from that trigger event.

¹ All emphases to quotations within this declaration have been added to the source material unless an indication to the contrary is provided.

41. Particularly relevant to Dr. Traynor's analysis of limitation [1.3], Dr. Traynor contends that requests from the application server to the network message server indicate "a corresponding one of the mobile end-user devices and one of a plurality of applications," on the basis that "Houghton discloses that command messages 'initiate a mobile terminal client trigger event in a mobile application 705 from a plurality of such applications' and that '[e]ach mobile application 205 has a fixed IP port number such as TCP/IP or UDP/IP' used to route the message to the application." Ex. 1003 ¶46 (alleging that the IP address / port number information indicates "one of a plurality of applications" and "a corresponding one of the mobile end-user devices").

42. With that background of Dr. Traynor's mapping of the prior art to claim 1, it is my opinion that Houghton does not disclose that the alleged requests sent from the "application servers" to the "network message server" (i.e., trigger events sent by Houghton's application server to Houghton's push server) indicate both (1) "a corresponding one of the mobile end-user devices" and (2) "one of a plurality of applications." As noted above, Dr. Traynor's sole theory as to why these trigger events indicate those requirements is that the *command messages sent from the push server* (i.e., the alleged "network message server") to the device agent would include IP address and port number information, which would allegedly indicate both "a corresponding one of the mobile end-user devices" and

“one of a plurality of applications.” But this says nothing of what the *trigger events* (i.e., the alleged “requests”) *received by* the push server (alleged “network message server”) would indicate. Simply put, the fact that a command message sent from a push server to a specific IP address and port number does not require (or even imply) that the trigger event sent from application server provided an IP address and port number to the push server.

43. Specifically, a POSITA would understand that in order for the push server to communicate with a particular device and application, the application server need not provide an IP address and port number. Instead, a POSITA would understand that because the push server itself (rather than the application server) maintains a “persistent managed, tested, and configured data connection... between push server 401/801 and push client 405/804” (Ex. 1005 at 23:4-9), it would logically be the *push server* responsible for determining the IP address and port number for any given communication were up to date. This is especially true given that a POSITA would readily understand that an application server may provide a trigger event to a push server significantly *before* the push server sends a corresponding message to one or more mobile devices. *See, e.g.*, Ex. 1003 ¶¶82-83 (alleging that “Houghton’s system would *collect and buffer messages*,” e.g., “when a mobile terminal does not have network connectivity”). If a mobile terminal does not have network connectivity (and thus has no available IP

address/port number combination) when a trigger event is sent from the application server to the push server, it would make little sense for the application server to provide an IP address / port number combination to the push server. Rather, a POSITA would understand that the push server would be responsible for determining which device and application should be contacted (including by determining an appropriate IP address and port number shortly before the command message from the push server is ultimately sent to the mobile terminal).

44. A POSITA's understanding that the push server (rather than the application server) is ultimately responsible for determining the appropriate devices and applications (including by determining appropriate IP addresses and port numbers) is supported by the Munson reference to which Dr. Traynor cites. In particular, Munson teaches that application service providers do **not** need to instruct a push server how to deliver push messages to a client device—the “CPS [Content Push Service] will deal with all the details of getting the content to a client device.” Ex. 1007, ¶[0040]. Munson additionally teaches that application servers can “push content to a group simply by naming the group,” and that they “do not need to specify each member of a group” (much less an IP address and port number for each member of a group) “for each push.” Ex. 1007, ¶[0043]. Thus, while I do not understand Dr. Traynor to allege that limitation [1.3] is inherently disclosed by a system in which an application server issues message requests to a

push server, the Munson reference shows that limitation [1.3] is *not* inherent in such a system.

VII. ANALYSIS OF DR. TRAYNOR’S GROUND 2

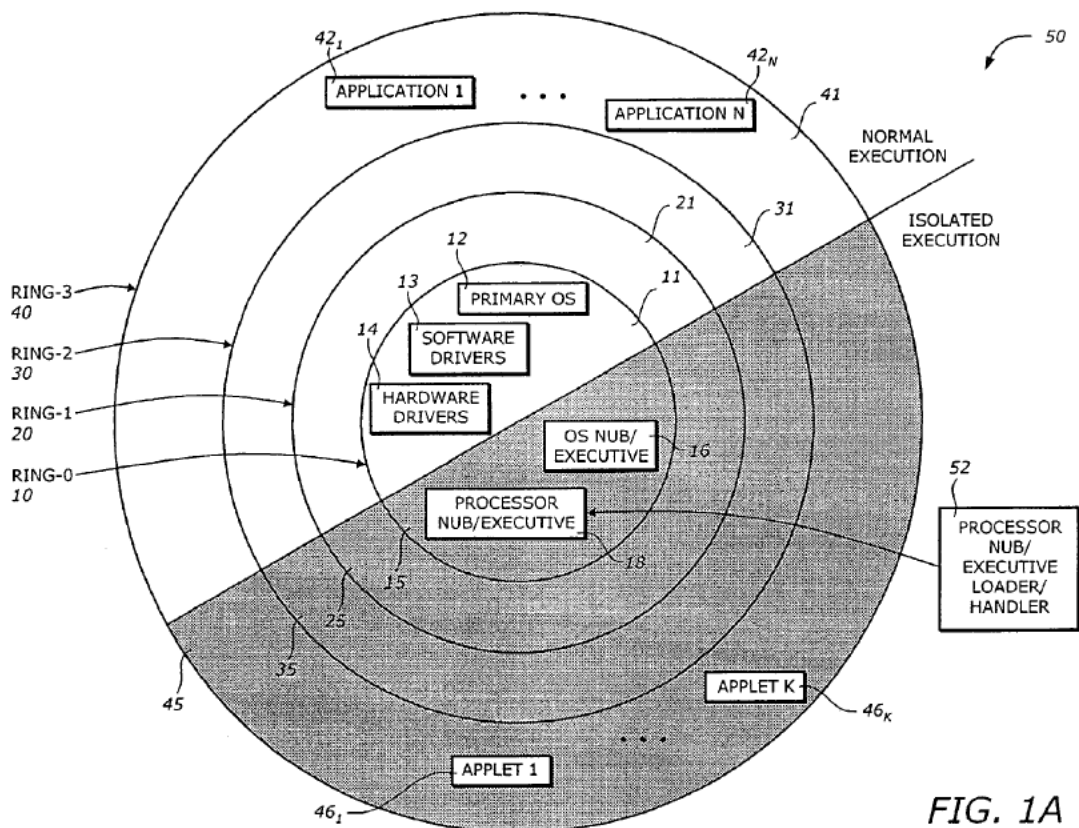
45. Limitation [1.7] of claim 1 recites that “for each received message,” each device messaging agent is configured to “*forward the application data in the message to [a] software process via a secure interprocess communication service.*” I disagree with Dr. Traynor that this limitation is disclosed or rendered obvious by Dr. Traynor’s invalidity combination for Ground 2A.

46. Although Dr. Traynor relies on a combination of Lee with Anderson and Ellison for his Ground 2A invalidity theory, he relies solely on elements of Ellison as disclosing the “secure interprocess communication service” portion of limitation [1.7]. In particular, Dr. Traynor relies on Ellison’s “isolated area 70,” which is “a memory area that is defined by [a processor] when operating in [an] isolated execution mode” (and which the Petition alternately alleges to be Ellison’s “secure platform”) as being a “secure interprocess communication service.” *See* Ex. 1003 ¶136.

47. According to the requirements of claim limitation [1.7], a “secure interprocess communication service” must be configured, for each of the plurality of received messages, to forward data to an application (i.e., “a software process corresponding to [an] application identifier”). Ellison’s “isolated area 70” cannot

meet that requirement; to the contrary, a POSITA would understand that Ellison's "isolated area 70" is unable to transfer data to *any* application because, as Ellison explicitly teaches, *no* applications are allowed to access isolated memory area 70.

48. In particular, Ellison teaches that every application within its system (Application 1 through Application N) resides in the ring-3 "normal execution" zone. Ex. 1013 at 4:26-29 ("The *normal execution ring-3, including applications 42_i to 42_N*, can access only ... the application pages 82. Both the normal execution ring-0 11 and *ring-3 41, however, cannot access the isolated area 70.*"). Ellison's Figure 1A (reproduced below) depicts each of its applications 42_i through 42_N residing in normal execution ring-3, which "cannot access the isolated area 70":



49. Ex. 1013, FIG. 1A; 4:26-29.

50. It is also my opinion that Ellison’s “secure platform” is not used to forward application data to applications. Instead, a POSITA would have recognized that Ellison’s secure platform is simply a system to monitor against unauthorized data accesses, not a platform for secure communication / forwarding of data, analogous to how antivirus software monitors malicious activity such as unauthorized changes to application behavior. Specifically, the “secure platform” depicted in Ellison’s Figure 2 (reproduced below) “includes the OS nub 16, the processor nub 18, a key generator 240, a hashing function 220, and a usage

protector 250, all operating within the isolated execution environment, as well as a software environment 210 that may exist either inside or outside the isolated execution environment.” See Ex. 1013 at 8:22-32. “[S]oftware environment 210 may include a plurality of subsets (e.g., subset 230).” Ex. 1013 at 9:28-29.

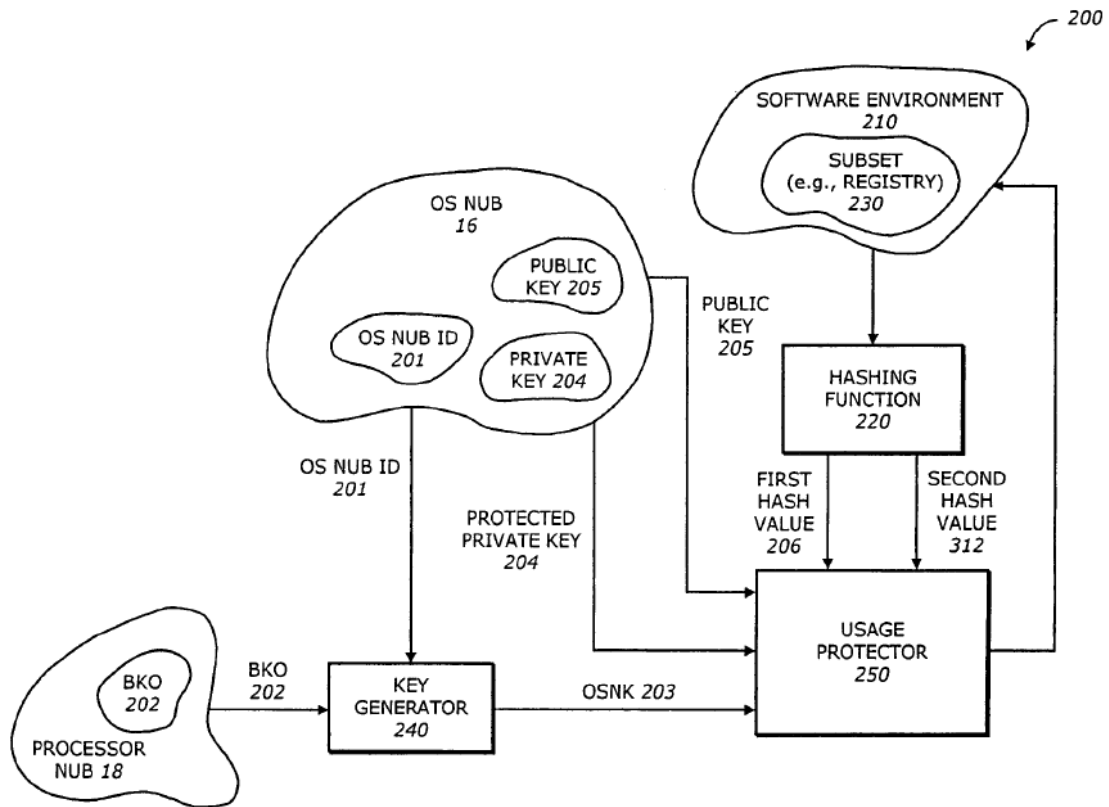


FIG. 2

51. Ex. 1013, FIG 2.

52. The purpose of this “secure platform” is to allow usage protector 250 to protect software environment 210 and/or subset 230 from intrusion or modification. See Ex. 1013 at 9:25-32 (“The usage the software environment 210 or the usage of the subset 230 is protected by the usage protector 250.”). In

particular, Ellison explains that the usage protector protects the usage of software environment 210 / subset 230 by taking one hash value of subset 230 at a first time, taking a second hash value of subset 230 at a second time, and comparing the hash values to see if there has been an unauthorized access to subset 230 in the duration of time between the hash values were created:

The subset 230 is hashed by the hashing function 220 to produce a first hash value 206 and a second hash value 312. One way to detect intrusion or modification of the subset 230 is to compare the state of the subset before and after a time period. The first and second hash values 206 and 312 are typically generated at different times and/or at different places.

The usage protector 250 is coupled to the key generator 240 to protect usage of the software environment 210 or the subset 230, using the OSNK 203. The usage protection includes protection against unauthorized reads, and detection of intrusion, tampering or unauthorized modification. ***If the two hash values are not the same, then the usage protector 250 knows that there is a change in the subset 230.*** If it is known that this change is authorized and an updated hash value has been provided, the usage protector 250 would merely report the result. Otherwise, the usage protector 250 may generate an error or a fault function. ***When a user is notified of the error, fault condition, he or she would know that the subset 230 has been tampered, modified. The user may take appropriate action. Therefore, the usage of the subset 230 is protected.***

53. Ex. 1013, 9:40-62. And although Ellison does state that there are “several different embodiments of the usage protector 250” within the “secure platform” represented by FIG. 2, these embodiments do not fundamentally change the principle of operation, and differ only regarding the specific mechanisms by which “usage protector 250 decrypts the subset” 230. *Id.* at 9:63-10:3 (stating that decryption may be accomplished using some combination of the OSNK 203, hash values 206 and 312, and the protected private key 204 and public key 205).

54. Accordingly, a POSITA would have recognized that the “secure platform” of Ellison’s Figure 2 is not a “communication service,” but is instead a mechanism by which the system can identify unwanted system changes. And I am aware of no evidence that Ellison’s “secure platform” is used to **forward** data at all, and Dr. Traynor cites to no such evidence. To the contrary, it is my opinion that a POSITA would have understood Ellison’s secure platform to be used to check for suspicious differences in system operation **after** data from Lee’s push message system had already been forwarded, rather than something that would constitute a secure communication service.

55. Indeed, I do not understand Dr. Traynor to allege anything to the contrary. Rather than alleging that Ellison’s secure platform is used to forward application data to applications, his declaration alleges only that Ellison’s secure platform would have protected the **applications** from malware, rather than

providing security via a communications service through which the malware would be **forwarded** (as the limitation [1.7] would require under Petition's theory). *See* Ex. 1003 ¶137 ("Accordingly, Ellison's secure platform would have provided protection for applications receiving push messages in Lee's push message system (e.g., malware contained within a push message would be detected using the usage protector 250's hashing function).").

56. Dr. Traynor's proposal, while it would provide a certain type of security protection, does not constitute a secure interprocess communication service that is used to forward application data from a push agent to an application. Instead, Dr. Traynor's proposal would only protect the application from potential security risks that might occur **after** the forwarded application data had already been forwarded and received by the application. For example, although Ellison's secure platform does not involve encrypting or securing communications sent over a communication bus, it could potentially identify malicious application behavior that might ultimately result from unsecured communications (e.g., malware) being sent over a communication bus to an application.

57. Ultimately, the distinction a POSITA would recognize between Dr. Traynor's solution and the requirement of limitation [1.7] of the '117 Patent is a distinction between the security of a communication service (as required by limitation [1.7]), as opposed to the security of a system component that sends or

receives data over a communication service (i.e., the security of an application that might receive a communication over an unspecified communication service, which is what the Ellison reference describes). As a general matter, a POSITA would have recognized that security provided beyond the endpoint of a communication service (e.g., at an element that sends or receives data via the communication service) does not transform an otherwise unsecured communication service into a secure communication service. For example, I might download a file to my laptop over an unsecured, public Wi-Fi connection; that unsecured Wi-Fi connection would not be a “secure communication service” used to download the file. And that unsecured Wi-Fi connection would not be transformed into a “secure communication service” even if I were to install antivirus software on my laptop. While the antivirus software might ultimately protect my laptop against the negative effects of receiving files over an unsecure connection (e.g., it might alert me to malware that I downloaded over the unsecure connection), that does not turn the Wi-Fi connection itself into a secure connection or secure communication service. Likewise, a POSITA would recognize that protecting an application from the potentially adverse effects of receiving malicious data over an unsecured system bus would not transform that unsecured system bus into a “secure interprocess communication service.”

58. Dr. Traynor’s declaration provides no reason to deviate from this ordinary understanding of what it means for data to be forwarded via a “secure interprocess communication service.” I first note that Dr. Traynor’s declaration provides no actual claim construction for “secure interprocess communication service” or what it means to “forward the application data... via a secure interprocess communication service.”

59. And while Dr. Traynor does quote an excerpt from the ’117 Patent relating to an “interprocess software communication bus” and states that it is relevant to the interpretation of “secure interprocess communication service,” he does not explain why he believes that portion of the specification supports his theory of invalidity. *See* Ex. 1003 ¶135 (quoting Ex. 1001 42:45-43:4).

60. Assuming that Dr. Traynor’s citation and quotation to the ’117 Patent is an attempt to provide some implicit definition or construction of “secure interprocess communication service,” I do not believe that the quoted portion of the ’117 Patent supports Dr. Traynor’s invalidity theory. In fact, it seems to show that the alleged “secure interprocess communication service[s]” in Ellison (i.e., “isolated area 70” and Ellison’s “secure platform”) are distinct from the invention of the ’117 Patent.

61. Most obviously, as Dr. Traynor states, the quoted portion of the ’117 Patent specification refers to an “interprocess software communication **bus**.” Ex.

1003 ¶135 (“While the ’117 patent does not explicitly define a ‘secure interprocess communication *service*,’ it does describe an ‘inter-process software communication *bus*.”). But neither Ellison’s “isolated area 70” nor its “secure platform” are a communication bus, so Dr. Traynor’s implication that a “secure interprocess communication service” is the same as a secure “interprocess software communication bus” is directly contrary to his Ground 2A theory of invalidity.

62. The specific teachings of the ’117 Patent relied on by Dr. Traynor likewise do not support his theory of invalidity. I have reproduced Dr. Traynor’s quotation of the ’117 Patent specification below (including the emphases added by Dr. Traynor and the omissions by Dr. Traynor):

In some embodiments of the agent communication bus 1630, the design of the agent communication bus depends on the nature of the design embodiments for the agents and/or other functions. **For example, if the agents are implemented largely or entirely in software, then the agent communication bus can be implemented as an inter-process software communication bus.** In some embodiments, such an inter-process software communication bus is a variant of D-bus ... or another inter-process communication protocol or system, running a session bus in which **all communications over the session bus can be secured, signed, encrypted or otherwise protected.** For example, the session bus can be further protected by **storing all software ... in secure memory, storing all software in encrypted form in secure memory, and/or executing all software**

and communications within a secure execution environment,
hardware environment and/or protected memory space.

63. Ex. 1003 ¶ 135 (emphases and omissions in original, quoting Ex. 1001, 42:45-43:4).

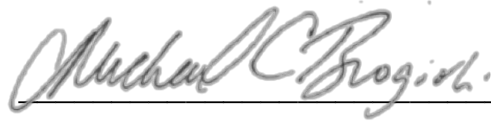
64. The above-quoted portion of the '117 Patent states that protection or security of a communication bus can be accomplished in various ways, none of which Dr. Traynor alleges would be present in Ellison (or any combination involving Ellison). For example, the '117 Patent implies that a secure bus can be implemented if “all communications over the session bus can be secured, signed, encrypted or otherwise protected.” Ex. 1001 at 42:51-58. But the teachings of Ellison that Dr. Traynor relies upon are not focused on the security of “communications” over the bus being used to transmit data, and are instead focused the security of endpoint applications that might ultimately receive the data. Dr. Traynor makes no showing that Ellison (or any combination involving Ellison) would involve securing, signing, encrypting, or otherwise protecting all communications over a bus. In fact, as I have alluded to above, Ellison appears to solve the problem of security in an entirely different manner than securing communications over the bus—Ellison is instead focused on identifying potentially malicious behavior within endpoint applications within a system, rather than creating a secure communication service within that system.

65. The portion of the '117 Patent on which Dr. Traynor relies goes on to state that “the session bus can be further protected by *storing all software... in secure memory*, storing all software in encrypted form in secure memory, *and/or executing all software and communications within a secure execution environment, hardware environment and/or protected memory space.*” Ex. 1001 at 42:58-64. But again, Ellison does not disclose or render obvious “storing *all software*” in secure memory (encrypted or otherwise), nor does it disclose or render obvious “executing *all software and communications* within a secure environment, hardware environment and/or protected memory space.” Instead, Ellison plainly teaches that a large portion of software is stored and executed *outside* of a secure execution environment. *See, e.g.*, Ex. 1013 at FIG 1A (showing all applications being executed within a “normal execution” environment, as opposed to other aspects of the system that execute in an “isolated execution” environment). Ellison’s isolated area and secure platform also do not relate to executing “communications” within a secure environment; rather, as I have mentioned above, a POSITA would understand that Ellison instead teaches protecting a software environment (such as an application), for example, *after* the application has already received an unsecure communication (such as a communication including malware).

66. Thus, it is my opinion that Ellison (either alone or in combination with the other references the Petition relies on for Ground 2A) does not disclose or render obvious the requirements of limitation [1.7].

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.

February 28, 2024

A handwritten signature in cursive script, reading "Michael C. Brogioli", written over a horizontal line.

Michael C. Brogioli, Ph.D.