

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM SERVICES LLC
and AMAZON WEB SERVICES, INC.,
Petitioners,

v.

HEADWATER RESEARCH LLC,
Patent Owner.

Case IPR2026-00106
Patent 10,321,320

**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	7
C.	Anticipation	7
D.	Obviousness	7
IV.	PERSON OF ORDINARY SKILL IN THE ART	9
V.	OVERVIEW OF THE '320 PATENT	10
VI.	CLAIM CONSTRUCTION.....	13
A.	“Periodic Timer”	14
B.	“Message Delivery Trigger”	17
C.	“Interprocess communication service”	20
VII.	OVERVIEW OF TS-23.140	23
VIII.	OPINIONS REGARDING DR. TRAYNOR’S TS-23.140-BASED THEORIES (CLAIM 1)	27
A.	Under Dr. Traynor’s interpretation of “message delivery trigger,” receipt of every received message in TS-23.140 is a message delivery trigger.	27
B.	Dr. Traynor’s invalidity theory also fails under the plain meaning of “message delivery trigger.”	32
1.	Dr. Traynor’s alleged triggers (1) and (2) are the same message delivery trigger.	33
2.	TS-23.140 does not disclose a User Agent becoming reachable as a separate message delivery trigger (alleged trigger (3)).	36

3. Dr. Traynor’s alleged triggers (4)-(6) are not message delivery triggers disclosed by TS-23.140.	39
4. Dr. Traynor’s alleged trigger (7) is part of the reception of a valid retrieval request.	41
IX. OPINIONS REGARDING DR. TRAYNOR’S TS-23.140-BASED THEORIES FOR CLAIMS 3 AND 4	43
A. The “earliest desired time of delivery” is not the “expiration of a periodic timer.”	43
B. TS-23.140’s “periodic polling” does not satisfy claim 3.....	48
C. The proposed combination of TS-23.140 and Adamczyk does not render claims 3 and 4 obvious.....	52
X. OPINIONS REGARDING DR. TRAYNOR’S TS-23.140-BASED THEORY FOR CLAIM 17	59
A. Dr. Traynor fails to show it would be obvious to combine TS-23.140 with Fok in his proposed manner.....	59
B. Dr. Traynor does not establish that his proposal, even if obvious, satisfies the claim requirements.....	67
XI. OPINIONS REGARDING DR. TRAYNOR’S HOUGHTON/MUNSON-BASED THEORIES	70
A. Dr. Traynor does not show that, for at least some received network element messages, receipt is not a message delivery trigger.....	70
B. A POSITA would not have combined Houghton and Munson as Dr. Traynor proposes.	72
C. Claims 3 and 4: Houghton-Munson (with or without TS-23.140).	77
D. Claims 3 and 4: Houghton-Munson-Adamczyk.	84
E. Claim 17: Houghton-Munson-Fok.....	88
1. Dr. Traynor fails to show it would be obvious to combine Houghton-Munson with Fok in his proposed manner.	88

2. Dr. Traynor does not establish that his proposal, even if obvious, satisfies the claim requirements.	96
XII. CONCLUSION.....	99

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. 10,321,320 (“the ’320 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 \$850 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 ’320 patent (Ex. 1001), the ’320 patent’s prosecution history (Ex. 1002), the Petition, the declaration of Dr. Patrick Traynor submitted in this proceeding (Ex. 1003)—a single declaration in which Dr. Traynor addresses both the ’320 patent and U.S. Patent No. 9,615,192 (“the ’192 patent”), which is the subject of IPR2026-00088—the transcript of Dr. Traynor’s July 9, 2026 deposition concerning that declaration

(Ex. 2010), the transcript of Dr. Traynor's deposition in IPR2024-00010 (Ex. 2011), Dr. Traynor's supplemental declaration in IPR2024-00010 (Ex. 2012), the prior art references asserted in this proceeding—including TS-23.140 (Ex. 1004), Houghton (Ex. 1007), Munson (Ex. 1017), and Adamczyk (Ex. 1021)—and each other 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 2009. 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 have previously held the role of 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 20+ 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 '320 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 2009, contains more information on my background and experience, as well as the cases in which I have served as an expert witness over the past four years.

17. No part of my compensation is contingent upon the outcome of this proceeding. I have no other interests in this proceeding 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 the 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.

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. 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 person of ordinary skill in the art ("POSITA") relating to the subject matter of the '320 patent as of January 28, 2009 would have had "(1) at least a bachelor's degree in computer science, electrical engineering, or a related field, and (2) three to five years of experience in services and application implementation in communications networks," and that "[a]dditional graduate education could substitute for professional experience, and vice versa." Ex. 1003, ¶47 (addressing the level of ordinary skill "relating to the subject matter of the '320 or '192 Patent").

30. For purposes of this declaration, I apply the following level of ordinary skill, which I understand to be consistent with the lower bound of Dr. Traynor's proposal: a bachelor's degree in computer science, electrical engineering, or a related field, plus at least three years of experience in services and application implementation in communication networks. In my opinion, the three years of experience can be obtained in either a professional setting or a research setting, because the relevant knowledge and skills—understanding of

messaging architectures, communication protocols, and the interaction between mobile devices and network elements—are developed in both settings. My opinions in this declaration would be the same under Dr. Traynor’s formulation of the level of ordinary skill.

31. I am, and as of January 28, 2009 was, at least a person of ordinary skill in the art under either formulation.

V. OVERVIEW OF THE ’320 PATENT

32. At a high level, the ’320 patent describes and claims a networked system comprising both a network server system and the device link agents residing on a plurality of wireless end-user devices. The network server system receives messages from various network elements for delivery to software components identified in those messages, buffers the message content, and evaluates a plurality of message delivery triggers before delivering the buffered content to a given device over a secure Internet data message link. Claim 1 is the only independent claim challenged in this proceeding; every other challenged claim (claims 2-18) depends, directly or indirectly, from claim 1. Claim 1 recites:

[1pre] A networked system comprising:

[1a] i) a network server system including

[1b1] a link interface to maintain a respective secure Internet data message link between the link interface and a respective device link agent on each of a plurality of wireless end-user devices,

[1b2] each of the wireless end-user devices comprising multiple software components authorized to receive messages via the device link agent on that device;

[1c1] a network interface to receive messages from a plurality of network elements, for delivery to respective ones of the software components identified in the messages,

[1c2] each network element authorized to send messages via the link interface to one or more of the software components on one or more of the wireless end-user devices; and

[1d1] a message buffer system including a memory and logic,

[1d2] the memory to buffer content from the received network element messages for which delivery is requested to any of the wireless end-user devices,

[1d3] the logic to determine when one of a plurality of message delivery triggers for a given one of the wireless end-user devices has occurred, wherein for at least some of the received network element messages, the receipt of such a message by the message buffer system is not a message delivery trigger, and for at least one of the message delivery triggers, the trigger is an occurrence of an asynchronous event with time-critical messaging needs, and

[1d4] upon determining that one of the message delivery triggers has occurred for the given one of the wireless end-user devices, the logic further to supply one or more messages comprising the buffered content for the given one of the wireless end-user devices to the transport services stack for delivery on the secure message link maintained between the transport services stack and a device link agent on the given one of the wireless end-user devices; and

[1e1] ii) the device link agents on the respective wireless end-user devices, each of the device link agents configured to

[1e2] maintain the respective secure Internet data message link over a wireless network to the link interface,

[1e3] receive secure Internet data messages from the network server system over the respective secure Internet data message link, including messages collected from multiple ones of the network elements and messages corresponding to multiple ones of the software components authorized to receive messages via the device link agent on that respective device, wherein at least a first subset of the secure Internet data messages contain both a unique identifier for a corresponding one of the software agents and data to be consumed by that software component, the data supplied from a respective network element corresponding to that software component, and

[1e4] for software components that are authorized to access messages received via the device link agent, cause messages with a unique identifier corresponding to a given one of those software applications to be securely delivered to a software process corresponding to the given software component.

Ex. 1001, claim 1.

33. Because every challenged claim depends, directly or indirectly, from claim 1, each requirement of claim 1 applies to all of the challenged claims.

Relevant to the opinions I provide below, limitation [1d3] imposes at least two distinct requirements concerning message delivery triggers: the message buffer system of the network server system must include “logic to determine when one of a plurality of message delivery triggers for a given one of the wireless end-user devices has occurred,” and, additionally, “for at least some of the received network element messages, the receipt of such a message by the message buffer system is not a message delivery trigger.”

34. Also relevant to my opinions is claim 3, which depends from claim 1 and recites: “The networked system of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.” Ex. 1001, claim 3. Claim 4 depends from claim 3, and further requires that “the period of the timer is fractionally shorter than a maximum data message interval beyond which the Internet data message link to the given device is taken down.” Ex. 1001, claim 4.

35. The ’320 patent’s specification describes a variety of message delivery triggers, including, for example, “periodic timer trigger[s],” triggers based on “waiting until a certain amount of service usage or traffic usage has occurred,” triggers “responding to a service agent message” or “request,” and triggers “initiated by a verification error condition.” Ex. 1001, 70:6-13. As I explain below, the claims’ requirement of a “plurality of message delivery triggers”—at least some of which are not the mere receipt of a message—reflects this design, in which the network server system’s logic distinguishes among multiple different events that can each cause delivery of buffered content.

VI. CLAIM CONSTRUCTION

36. I understand that Dr. Traynor did not offer any formal claim constructions, and instead purported to apply the plain and ordinary meaning of the claim terms. For each claim limitation I discuss, I likewise apply what I believe would have been the plain and ordinary meaning to a POSITA in light of the

claims, the specification, and the file history of the '320 patent. However, as I explain in this section, certain of Dr. Traynor's opinions apply understandings of the terms "periodic timer" and "message delivery trigger" that are inconsistent with those terms' plain and ordinary meanings.

A. "Periodic Timer"

37. The term "periodic timer" appears in claim 3, which depends from claim 1 and further requires that "one of the message delivery triggers is the expiration of a periodic timer." While the '320 patent does not explicitly define "periodic timer," a POSITA would understand it to refer to a timer that recurs in accordance with a given time period. This is the ordinary usage of "periodic" in this field. For example, The Authoritative Dictionary of IEEE Standards Terms (7th Ed. 2000) defines "periodic" as "[o]f or pertaining to a series of specified waveforms or features which repeat or recur regularly... in time," as opposed to "aperiodic," in which the "waveforms or features repeat or recur... irregularly in time." Ex. 2013, 13.

38. In the context of messaging systems, a good example of something that occurs periodically is "periodic polling," which Dr. Traynor himself discusses. See Ex. 1003, ¶643 (acknowledging that "the periodic polling would be done at periodic intervals (i.e., after a certain time period has elapsed/expired) that repeat regularly"). A POSITA would understand that periodic polling is a scheme in

which a system will “poll” (or inquire as to the status of some value) in a recurring manner at some defined frequency—the time between these recurring polls is the “period” of the periodic polling.

39. This understanding of “periodic” is also consistent with the usage of “periodic” in patents assigned to Petitioners’ affiliate Amazon Technologies, Inc. For example, U.S. Patent No. 12,210,419, assigned to Amazon Technologies, Inc., distinguishes between performing a task “according to a published, but not necessarily periodic, schedule” (Ex. 2014, claim 3), thereby recognizing that following a set schedule is not necessarily “periodic.” In other words, in ordinary technical usage, what makes a timer “periodic” is that it repeats at a regular, recurring interval—not merely that it measures some period of time once. (I also note that a periodic *schedule* can be accomplished without a periodic *timer*; e.g., street sweeping Mondays at 8am occurs on a periodic schedule, but no periodic timer need be involved.)

40. As I discuss in later sections of this declaration, Dr. Traynor applies a far broader understanding of what constitutes a “periodic timer.” For example, Dr. Traynor opines that “specifying the desired time of delivery triggers a periodic timer upon the receipt of a message that expires when the delivery time is reached,” and that such a timer “would be periodic in that each time a message is received, the timer is triggered and continues for a particular period at the

expiration of which, the message is delivered to the intended recipient.” Ex. 1003, ¶641. At his deposition, Dr. Traynor confirmed that he applied a meaning of “periodic timer” in which the timer need only be “related to a time period,” such that a timer that runs exactly once for a single message could be a “periodic timer.” Ex. 2010, 28:20-29:14. And in a previous deposition, Dr. Traynor went so far as to state that “expiration of a periodic timer” might even be satisfied by a user looking at the clock and waiting until a particular time had passed before sending a message. Ex. 2011, 17:14-18:7.

41. In my opinion, Dr. Traynor’s understanding of “periodic timer” is inconsistent with the term’s ordinary meaning, and would have the effect of reading the word “periodic” out of claim 3 entirely. The phrase “expiration of a ... timer” already implies that some time period is being measured. See Ex. 2013, 12 (defining “arm a timer” as “[t]o start a timer measuring the passage of time, enabling the notification of a process when the specified time or time interval has passed”). Accordingly, if all that were required for a timer to be “periodic” were that the timer be “triggered and continue[] for a particular period” (Ex. 1003, ¶641), then every timer would be a “periodic timer,” because every timer that can “expire” necessarily expires after some period of time. Notably, at his deposition, Dr. Traynor was unable to identify any example of a timer that would not be “periodic” under his usage of that term. Ex. 2010, 31:1-16 (“And sitting here right

now, can you give me an example of a timer that is not relating to a time period?

THE WITNESS: ... I don't have any examples at the moment off the top of my head.”).

42. In my opinion, a POSITA reading claim 3 would understand that the claim's use of “periodic”—a word the applicant chose to include in addition to “timer”—requires a timer that recurs at regular intervals, consistent with the IEEE definition and with ordinary usage in the field.

B. “Message Delivery Trigger”

43. In my opinion, Dr. Traynor's application of “message delivery trigger” is inconsistent with what I understand its plain meaning to be, which is something that triggers message delivery—not merely something that is necessary for message delivery (i.e., not merely a condition that “must occur” at some point before message delivery can complete). In my opinion, a POSITA would understand a “message delivery trigger” to be a condition that is sufficient to set message delivery in motion in the normal sequence of events, rather than any arbitrary precondition of delivery.

44. An analogy illustrates the distinction. In order for a firearm to be discharged, numerous conditions must occur: the firearm must be constructed; it must be loaded with ammunition; the safety must be off; and the trigger must be pulled. No one, however, would refer to construction of the firearm, or loading of

the firearm, as a “gunfire trigger.” Everyone would instead understand that pulling the trigger of an operational, loaded firearm is what triggers the gunfire, because in the normal sequence of operation, that is the condition that sets the firing sequence in motion. The same distinction applies in messaging systems: many conditions are necessary predicates of a delivery (the message must exist, the server must be powered on, a network path must exist), but a POSITA would not call each such precondition a “message delivery trigger.”

45. This understanding is supported by the extrinsic evidence on which Dr. Traynor himself has relied. In IPR2024-00010, Dr. Traynor asserted that a “trigger” involves both a “condition and an associated action,” such that “[i]f the condition is satisfied, the rule is triggered.” Ex. 2012, ¶61. Applying that framing to a “message delivery trigger,” the associated action is message delivery: if the condition is satisfied, message delivery is triggered. That describes a sufficient condition—something whose occurrence sets delivery in motion—not any condition that merely “must occur” somewhere upstream of an eventual delivery.

46. By contrast, and as I discuss in detail below, Dr. Traynor has advanced an understanding under which a “message delivery trigger” includes any “condition[] imposed on the messages controlling whether a message will be delivered at all,” which he analogizes to “any traffic lights plus road restrictions, like maximum height/weight to get to a destination.” Ex. 1003, ¶601. In his

supplemental declaration in IPR2024-00010, he likewise described the “sequence of triggers that must occur before message delivery is initiated,” and disagreed with any attempt “to distinguish a message delivery trigger from a condition that must occur before [the] message is delivered.” Ex. 2010, ¶55; *id.*, 34 (heading F). In my opinion, that “must occur” framing is not the plain and ordinary meaning of “message delivery trigger”; it sweeps in every necessary precondition of delivery, no matter how far removed from the event that actually sets delivery in motion. At his deposition, Dr. Traynor confirmed that this broad reading is the single meaning he applies. Asked to describe the plain meaning of “message delivery trigger,” he directed counsel to paragraph 200 of his declaration—the paragraph in which he sets out, for the ’192 patent, the same seven alleged triggers and same three groupings that he sets out for the ’320 patent at Ex. 1003, ¶601—and testified that he would “rely on what I’ve written here to cover the plain and ordinary meaning of what a trigger is.” Ex. 2010, 8:5-8. He then testified that his “opinion in paragraph 200 is that a trigger encompasses all seven of these items.” Ex. 2010, 8:23-24. And when asked whether it was his opinion “that the claimed trigger must be a condition after which the next step is an attempt to deliver the message,” he answered: “So as I say in paragraph 200, the plain and ordinary meaning of ‘trigger’ encompasses all seven of these items. I’ve merely grouped them together logically, but I believe all seven qualify as triggers.” Ex. 2012, 9:5-11. In my

opinion, that testimony makes clear that the three “tiers” in his declaration are not alternative constructions he offers; he applies the broadest of them.

47. I have been asked, as part of rendering my opinions in this proceeding, to analyze Dr. Traynor’s invalidity theories under both his understanding of “message delivery trigger” and my own. In my opinion, the challenged claims are valid under either interpretation.

C. “Interprocess communication service”

48. In my opinion, an “interprocess communication service” is distinct from any communication between processes (i.e., it is distinct from “interprocess communication”), because it requires a *service* that is provided to the applications by some component.

49. An example of “interprocess communication” that does not involve an “interprocess communication service” is communication via shared memory, in which memory is provided by the operating system that may be shared between processes. This allows processes to read from and write to the shared memory, thus allowing them to communicate with one another. For example, process A may write a message to the shared memory that process B may then read from. But once the shared memory region is established by the operating system, the processes that use the shared memory for interprocess communication manage the communication themselves (including elements of the communication such as

timing, access control, and format of the exchanged data). Thus, the use of shared memory for communication between processes does not involve a “interprocess communication service”—it instead involves interprocess communication involving a mechanism (shared memory) wherein the actual interprocess communication is not provided via a service but is instead managed by individual applications.

50. In contrast, D-bus (as described in the ’320 patent) is an example of a system that “helps applications/agents talk to one another”; such a service defines how applications interact with one another and is thus a *service* for interprocess communications. *See* Ex. 1001, 43:24–31. By way of analogy, telephone service or mail service are communication services. But individuals may communicate with one another without using a communication service (e.g., I may simply walk to my neighbor’s house and speak with them, without using any communication service). What distinguishes an “interprocess communication *service*” from “interprocess communication” is whether a specific communication service is being provided that defines and manages communication protocols between processes.

51. This is consistent with the language of claim 17, which requires “causing messages to be securely delivered to a software process by initiating delivery of each such message *on* the secure interprocess communication service.” There thus must exist a specific “service” on which the messages are delivered,

and the claim limitation cannot be satisfied by secure delivery of messages, irrespective of whether such messages are delivered on a “secure... service” provided to the processes that are communicating between each other.

52. My opinions are also supported by peer-reviewed IEEE articles discussing interprocess communication services spanning 1998–2012. For example, in *Distributed IPC using Virtual Device Driver in Monolithic Kernel*, Susmit Bagchi in 2012 noted that “large scale distributed computing applications require the efficient and transparent distributed interprocess communication (IPC) *as a service* from the underlying operating systems.” Ex. 2015 at 51; *see also id.* at 52 (“The kernel-level distributed IPC architecture creates a system-wide virtual layer of service to user applications....”); *id.* at 53 (“The registration command is used by the application processes to register to VxD in order to use distributed IPC service of the kernel.”). This shows that an IPC *service* is provided to the applications that utilize it. Likewise, in *A Real-Time IPC Service Over ATM Networks for the Chorus Distributed System*, Christophe Lizzi and Eric Gressier-Soudan in 1998 described “a real-time interprocess communication service (RT-IPC) suitable for distributed systems built over ATM networks” which is “implemented in the Chorus distributed operating system, as servers running in the micro-kernel address space.” Ex. 2016 at 1013. The article makes clear that the “service provider” is not the entity *using* the service, because the service is

provided to the using entities. *See id. at 1014* (noting that the “IPC user will not overtake the traffic contract admitted by the service provider”); *id. at 1015* (introducing “a new IPC service for the Chorus system” which “support[s] the level of service expected by distributed hard real-time applications”); *id. at 1019* (discussing “the time the message is submitted to the RT-IPC service, until its reception by the destination RT-port,” making clear that the IPC service is limited to transferring data between endpoints, rather than specifying how applications manage and verify information after receiving it).

VII. OVERVIEW OF TS-23.140

53. TS-23.140 (Ex. 1004) is a 3GPP technical specification describing the functional requirements of the Multimedia Messaging Service (“MMS”). Dr. Traynor’s opinions map the claimed “network server system” onto TS-23.140’s MMS Relay/Server, and the claimed “device link agent” onto TS-23.140’s MMS User Agent. *See, e.g., Ex. 1003, ¶¶539, 545, 548.* As shown in Figure 2 of TS-23.140, the MMS Relay/Server sits between, on one side, various message sources (including MMS Value Added Service (“VAS”) Applications and External Servers), and, on the other side, the MMS User Agents, which are agents operating on wireless end-user devices (UEs/mobile devices). Ex. 1004, 17 (FIG. 2).

54. Before setting out my specific opinions, it is helpful to describe how message delivery actually works in TS-23.140, because several of Dr. Traynor’s

opinions depend on characterizations of TS-23.140 that, in my opinion, are not accurate.

55. As an initial matter, I note that TS-23.140, like other 3GPP specifications, uses the word “shall” to denote a mandatory requirement of an implementation of the specification, as opposed to “may,” which refers to something that may optionally be implemented. A POSITA reading TS-23.140 would understand this convention, which is standard in technical specifications of this kind.

56. TS-23.140 specifies that the “MMS Relay/Server shall provide” specified functionalities, including “receiving and sending MM [multimedia messages].”

The MMS Relay/Server shall provide the following functionalities:

- receiving and sending MM;
- conversion of messages arriving at the recipient MMS Relay/Server from legacy messaging systems to MM format (e.g. facsimile to MM) if interworking with legacy messaging systems (MM3) is supported;
- conversion of MMs leaving the originator MMS Relay/Server to legacy messaging systems to the appropriate message format (e.g. MM to internet email) if interworking with legacy messaging systems (MM3) is supported;
- message content retrieval;
- MM notification to the MMS User Agent;

Ex. 1004, 21. Although the MMS Relay/Server is the entity responsible for physically “sending [the] MM” to the MMS User Agent, TS-23.140 specifies that it is the MMS User Agent that is tasked with initiating MM delivery: among the “application layer functionalities” that “[t]he MMS User Agent shall provide” is

“the retrieval of MMs (initiate MM delivery to the MMS User Agent).” Ex. 1004, 19.

57. TS-23.140 describes how the MMS User Agent initiates MM delivery. Whenever a recipient MMS Relay/Server receives an MM, it “shall generate a notification to the recipient MMS User Agent.” Ex. 1004, 28. “In a response to the notification the MMS User Agent” has two options: reject the MM, or “retrieve the MM, either immediately or at a later time (i.e. defer the MM retrieval), either manually or automatically, as possibly determined by the operator configuration and user profile.”

In a response to the notification the MMS User Agent shall be able to

- reject the MM or
- retrieve the MM, either immediately or at a later time (i.e. defer the MM retrieval), either manually or automatically, as possibly determined by the operator configuration and user profile.

Ex. 1004, 28.

58. Retrieval occurs in specific response to a retrieval request from the MMS User Agent: “The recipient MMS User Agent shall be able to request retrieval of an MM from the recipient MMS Relay/Server based on the Message Reference received in a notification.” Ex. 1004, 29. When the MMS User Agent provides a valid retrieval request (i.e., a retrieval request conforming to TS-23.140’s requirements, including being within the message expiry period and satisfying any size restriction), the MMS Relay/Server must initiate delivery of the

MM: “Upon retrieval request the recipient MMS Relay/Server ... shall deliver the MM to the recipient MMS User Agent.”

7.1.3 Retrieval of a Multimedia Message in the recipient MMSE

The recipient MMS User Agent shall be able to request retrieval of an MM from the recipient MMS Relay/Server based on the Message Reference received in a notification. If MMBoxes are supported, the MMS User Agent shall be able to request retrieval of an MM from the user’s MMBox, based on a Message Reference received from a previous MMBox operation.

Within a retrieval request the recipient MMS User Agent may indicate a size restriction of the returned MM (i.e., maximum size) that the MMS Relay/Server is to use in processing the retrieval request.

Upon retrieval request the recipient MMS Relay/Server

- shall deliver the MM to the recipient MMS User Agent

Ex. 1004, 29 (annotated).

59. The only exception to this requirement that TS-23.140 identifies is that the MMS Relay/Server “should not deliver the MM (or any adaptation of the MM) to the recipient MMS User Agent unless the size restriction set by the MMS User Agent is met.” Ex. 1004, 29. In other words, so long as a retrieval request does not specify a size restriction inconsistent with the requested, non-expired message, the retrieval request shall initiate delivery of the message to the MMS User Agent. This is the only condition that TS-23.140 discloses that “shall” initiate delivery of an MM to the MMS User Agent.

60. At his deposition, Dr. Traynor was unable to identify any instance in TS-23.140 in which the MMS Relay/Server “shall deliver the MM to the recipient MMS User Agent” other than upon a retrieval request. Ex. 2010, 9:21-10:23. In my opinion, that is because there is none: in TS-23.140, reception of a valid retrieval

request—and only reception of a valid retrieval request—is disclosed as initiating the MMS Relay/Server’s delivery procedure toward the MMS User Agent.

61. Finally, I note that in TS-23.140, for every received message, receipt of the message by the MMS Relay/Server (including its memory store/MMBox) is a necessary predicate for that message ever to be delivered to an MMS User Agent: the MMS Relay/Server cannot generate a notification for, and cannot deliver, a message it has not received. Ex. 1004, 28 (“Upon reception of an MM the recipient MMS Relay/Server ... shall store the MM at least until ... the MM is delivered ...[and] shall generate a notification to the recipient MMS User Agent.”). This point is significant to my opinions below regarding the claim requirement that, “for at least some of the received network element messages, the receipt of such a message by the message buffer system is not a message delivery trigger.”

VIII. OPINIONS REGARDING DR. TRAYNOR’S TS-23.140-BASED THEORIES (CLAIM 1)

A. Under Dr. Traynor’s interpretation of “message delivery trigger,” receipt of every received message in TS-23.140 is a message delivery trigger.

62. As noted above, the challenged claims impose two distinct requirements concerning message delivery triggers: the network server system must have “logic to determine when one of a plurality of message delivery triggers ... has occurred,” and, “for at least some of the received network element messages, the receipt of such a message by the message buffer system is not a

message delivery trigger.” Ex. 1001, claim 1[d3]. As I explained in my overview of TS-23.140, the reference discloses only one condition upon which the MMS Relay/Server “shall” initiate delivery of a message to an MMS User Agent: reception of a valid retrieval request. In order to arrive at a “plurality” of message delivery triggers, Dr. Traynor applies an extraordinarily broad interpretation of “message delivery trigger,” organized into three progressively broader “tiers”: (i) “a condition after which the next step is an attempt to deliver the message (e.g., the last traffic light before a destination)”; (ii) “conditions that start[] or restart[] an attempted delivery (e.g., any traffic light on the way to a destination)”; and (iii) “conditions imposed on the messages controlling whether a message will be delivered at all (e.g., any traffic lights plus road restrictions, like maximum height/weight to get to a destination).” Ex. 1003, ¶601.

63. Dr. Traynor has made clear that, in his view, a “message delivery trigger” includes any condition that “must occur” before message delivery. In IPR2024-00010, he described “a sequence of triggers that must occur before message delivery is initiated,” and opined that any event that “initiates a message delivery sequence culminating with the delivery of the message” “constitute[s] a message delivery trigger.” Ex. 2012, ¶¶51-58. Applying that view, he opined that because “a message cannot be retrieved without first receiving the MM notification and without the corresponding inclusion of the message reference,” the MM

notification itself constitutes a “message delivery trigger”—even though the notification occurs “before the message retrieval requests.” Ex. 2012, ¶¶52-53, 58. He also expressly disagreed with any attempt “to distinguish a message delivery trigger from a condition that must occur before [the] message is delivered.” Ex. 2012, 34 (heading F).

64. In TS-23.140, for every received network element message, receipt of the message by the MMS Relay/Server’s memory store/MMBox is necessary for that message ever to be delivered to an MMS User Agent: the MMS Relay/Server stores each received MM at least until it is delivered, expires, or is rejected, and it is the reception of the MM that causes the MMS Relay/Server to generate the notification that begins the delivery sequence. Ex. 1004, 28. Accordingly, under Dr. Traynor’s own framework, receipt of each message is a “condition[] ... controlling whether a message will be delivered at all” (Ex. 1003, ¶601); it is one of the first “traffic light[s] on the way to [the] destination” of eventual message delivery (*id.*); and it is “a condition that must occur before [the] message is delivered” (Ex. 2012, 34). Indeed, receipt of the message is the very event that “initiates [the] message delivery sequence culminating with the delivery of the message” (Ex. 2012, ¶58), because it is what causes the notification to be generated.

65. Dr. Traynor’s declaration in this proceeding does not actually allege that, for at least some received network element messages, message receipt is not a message delivery trigger. Instead, he opines that “for at least some received network element messages, message receipt by the memory store/MMBox *alone* is not a trigger.” Ex. 1003, ¶603 (emphasis added). In my opinion, the claim language is materially different from Dr. Traynor’s restatement of it. The claims require that, for at least some received network element messages, “the receipt of such a message by the message buffer system is not a message delivery trigger”—without the qualifier “alone.” A showing that message receipt is not the only trigger associated with a message is not a showing that message receipt is not a trigger. Under Dr. Traynor’s expansive interpretation of “message delivery trigger,” the receipt of every message qualifies as a message delivery trigger, and the negative limitation is therefore not satisfied.^[P]_{SEP}] I understand that Dr. Traynor also states, in the alternative, that “if a trigger is a condition after which the immediate next step is an attempt to deliver the message (conditions (1)-(3)), the receipt of the MM—with or without a desired time of delivery—is not a trigger,” such that “none of the received messages are triggers.” Ex. 1003, ¶603. In my opinion, that alternative does not reflect the interpretation Dr. Traynor actually applies. As I noted above, he testified that the plain and ordinary meaning of “trigger” “encompasses all seven of these items,” that he “merely grouped them together logically,” and that

“all seven qualify as triggers.” Ex. 2010, 9:8-11. Under the interpretation he applies—the same interpretation under which he identifies the “plurality” of message delivery triggers that the claims require—the receipt of every received network element message in TS-23.140 is a message delivery trigger.

66. Dr. Traynor’s only specific example of a message whose receipt would purportedly not be a trigger is “the receipt of message with a specified time of delivery,” where, in his view, “attempted delivery is only triggered by a determination that the earliest desired time of delivery condition is met.” Ex. 1003, ¶603. This does not change my opinion, for two reasons. First, TS-23.140 does not disclose any received message that specifies a time at which delivery to the MMS User Agent occurs; the originator may specify only an earliest desired time of delivery, after which the ordinary notification-and-retrieval sequence still must run its course before any delivery occurs—the MMS User Agent must still receive a notification and must still issue a valid retrieval request. Ex. 1004, 27-29; *see also* Ex. 2012, ¶¶52-53 (Dr. Traynor acknowledging that “when the MMS Relay/Server receives a message, it generates and sends a notification ... that includes a message reference corresponding to the message,” and that “a message cannot be retrieved without ... the message reference”). Second, even for a message with an earliest desired time of delivery, the initial reception of the message remains “a condition that controls whether an attempted message delivery is made” (Ex. 1003, ¶603), a

“traffic light on the way to a destination” (id., ¶601), and “a condition that must occur before [the] message is delivered” (Ex. 2012, 34)—all formulations that Dr. Traynor has said satisfy the term “message delivery trigger.” Accordingly, in my opinion, under Dr. Traynor’s own proffered interpretation, receipt of every received message in TS-23.140—including messages specifying an earliest desired time of delivery—constitutes a “message delivery trigger,” and TS-23.140 therefore does not satisfy the requirement that “for at least some of the received network element messages, the receipt of such a message by the message buffer system is not a message delivery trigger.”

B. Dr. Traynor’s invalidity theory also fails under the plain meaning of “message delivery trigger.”

67. As noted above, I have a different understanding of “message delivery trigger” than Dr. Traynor does. I explain below why Dr. Traynor’s Ground 1 theory additionally fails under my understanding of “message delivery trigger,” because the claims require “a plurality of message delivery triggers.” In addressing his alleged triggers below, I use the three groupings that appear in his declaration for ease of reference. As explained above, I do not understand those groupings to be alternative constructions that Dr. Traynor offers; he testified that all seven of his alleged triggers qualify under the single plain and ordinary meaning he applies. Ex. 2010, 9:8-11.

1. Dr. Traynor's alleged triggers (1) and (2) are the same message delivery trigger.

68. Under the narrowest of his three groupings—"a condition after which the next step is an attempt to deliver the message"—Dr. Traynor identifies three alleged triggers in TS-23.140:

(1) When the recipient User Agent is configured for manual retrieval, the User Agent manually issues (trigger) a retrieval request in response to a notification to initiate a delivery attempt. Id., 28 (§7.1.2); id., 28-29 (§7.1.2.1).

(2) When the recipient User Agent is configured for automatic delivery, the User Agent automatically issues (trigger) a retrieval request in response to a notification to initiate a delivery attempt. Id.

(3) When the MMS User Agent is unavailable to receive a message, an indication (trigger) that the MMS User Agent has later become available/reachable (e.g., moves into coverage, switches User Agent on) to initiate a delivery attempt. Id., 29 (§7.1.3).

Ex. 1003, ¶601.

69. In my opinion, alleged triggers (1) and (2) are the same message delivery trigger, initiated in different ways on the device. The claims require that the network server system have "logic to determine when one of a plurality of message delivery triggers ... has occurred." Ex. 1001, 167:28-30. The claims thus view the message delivery triggers from the vantage point of the network server system: the triggers are events whose occurrence the server's own logic determines. Whether Dr. Traynor has identified two message delivery triggers, or one trigger initiated in two different ways, must accordingly be evaluated from the

perspective of the alleged network server system (the MMS Relay/Server). From that perspective, alleged triggers (1) and (2) are the same event: reception of a retrieval request. The retrieval request is the same message in both cases—MM1_retrieve.REQ—and its only defined information element is the mandatory “Message Reference” identifying the MM to be retrieved. Ex. 1004, 71 (Table 11). Nothing in the request conveys to the MMS Relay/Server whether the recipient User Agent issued it manually or automatically; that distinction reflects nothing more than the retrieval mode configured on the device. See Ex. 1004, 29 (“The retrieval mode employed by the recipient MMS User Agent for a particular MM may be based either on the user settings in the terminal or on the recommendation carried in the MM notification.”). And the MMS Relay/Server responds identically in either case: “[u]pon retrieval request,” it “shall deliver the MM to the recipient MMS User Agent.” Ex. 1004, 29. An event that the alleged network server system receives in one and only one form—reception of a retrieval request—is one message delivery trigger, not two.

70. In my opinion, counting the same server-side event as multiple “triggers” based on differences that are invisible to the server would read the claimed “plurality” out of the claims. Under Dr. Traynor’s counting method, any system responsive to a single message delivery trigger would satisfy the “plurality of message delivery triggers” limitation, because the same trigger can always arise

under some arguably different device-side circumstance: the same retrieval request might be issued because the user pressed a button or because the device was configured to issue it, at 9:01 a.m. or at 10:02 a.m., in response to one notification or another. If each such variation gave rise to a new “message delivery trigger,” the word “plurality” would do no work at all. A POSITA would instead understand the claims to require at least two different events that the server’s logic can determine to have occurred—not the same event counted twice based on device-side details the server never sees.

71. This understanding is consistent with limitation [1d4], which requires that “upon determining that one of the message delivery triggers has occurred for the given one of the wireless end-user devices,” the logic supplies “one or more messages comprising the buffered content ... to the transport services stack for delivery.” Ex. 1001, claim 1 [1d4]. In other words, once the server’s logic determines that one of the message delivery triggers has occurred, delivery is attempted by placing the messages for delivery on the secure link. The claim structure thus confirms that the “triggers” are events the server’s logic detects and acts upon—reinforcing that trigger identity is assessed from the server’s perspective.

72. I further note that Dr. Traynor’s treatment of alleged triggers (1) and (2) is inconsistent with the reasoning he applies elsewhere. In explaining why

message receipt purportedly does not satisfy the negative limitation, Dr. Traynor reasons that message receipt “*alone* is not a trigger” because other conditions must also be satisfied. Ex. 1003, ¶603 (emphasis added). But precisely the same is true of a retrieval request in TS-23.140: even after a retrieval request is received, delivery will not complete if the User Agent is or becomes unreachable (per his alleged trigger (3)), or if a size restriction in the request is not satisfied (per his alleged trigger (7)). Ex. 1004, 29. If Dr. Traynor’s “alone is not a trigger” reasoning could disqualify message receipt, it would equally disqualify the retrieval request. This illustrates, in my opinion, that Dr. Traynor’s framework is not being applied consistently, but is instead being adjusted limitation-by-limitation to reach the claimed elements.

2. TS-23.140 does not disclose a User Agent becoming reachable as a separate message delivery trigger (alleged trigger (3)).

73. Dr. Traynor’s alleged trigger (3) is “an indication (trigger) that the MMS User Agent has later become available/reachable (e.g., moves into coverage, switches User Agent on).” Ex. 1003, ¶601. In my opinion, this is not a message delivery trigger disclosed by TS-23.140; User Agent availability is part of the process by which an already-triggered delivery completes. The passage of TS-23.140 on which this allegation rests appears within the list of actions the MMS Relay/Server takes “[u]pon retrieval request”: the MMS Relay/Server “shall store

messages in the network until the recipient MMS User Agent becomes reachable (e.g. user moves back into coverage, switches MMS User Agent on) or until the MM expires.” Ex. 1004, 29. In other words, this buffering behavior is part of how message delivery—which is triggered “[u]pon retrieval request”—functions: once delivery is triggered, the outgoing message is buffered until delivery completes or the message expires.

74. Such instances of unreachability during delivery would be understood by a POSITA to be uncommon in operation, because message delivery immediately follows the reception of a valid retrieval request, and the MMS User Agent must have been available on the network moments earlier in order to transmit that request. More fundamentally, TS-23.140 discloses the “becomes reachable” language as an indication of how long the message will be buffered—not as a separate message delivery trigger. Indeed, if TS-23.140’s disclosure of the duration of message buffering were a message delivery trigger, then the other endpoint of the same sentence—“until the MM expires”—would equally be a “message delivery trigger,” which would make no sense: message expiry ends the possibility of delivery rather than triggering it. I understand that Dr. Traynor does not contend in this proceeding that MM expiry is a message delivery trigger.

75. Dr. Traynor’s alleged trigger (3) fails for an additional reason: neither his declaration nor TS-23.140 identifies what the “indication” that the MMS User

Agent has become reachable would actually be, as distinct from a retrieval request. *See* Ex. 1003, ¶601 (asserting that there would be “an indication (trigger) that the MMS User Agent has later become available/reachable,” without identifying the indication). To the extent the User Agent becoming available or reachable actually triggers a delivery attempt, a POSITA implementing TS-23.140 would understand that this indication would simply be an additional retrieval request: a retrieval request signals to the MMS Relay/Server that the User Agent is available, and implementing the indication of availability in this manner ensures that the User Agent is not only available but also ready and willing to receive the requested message.

76. In short, the User Agent becoming available or reachable is a requirement for completing a delivery that has already been triggered. To the extent it is treated as a trigger for a delivery attempt, nothing in TS-23.140 or Dr. Traynor’s declaration shows that it would be implemented by anything other than a message retrieval request—i.e., the same message delivery trigger as alleged triggers (1) and (2). And here too, Dr. Traynor’s own “alone is not a trigger” reasoning (Ex. 1003, ¶603) would apply with equal force: in TS-23.140, User Agent availability “alone” does not result in delivery; among other things, the User Agent must have requested delivery of the message.

3. Dr. Traynor's alleged triggers (4)-(6) are not message delivery triggers disclosed by TS-23.140.

77. Under his second, broader tier—"conditions that start[] or restart[] an attempted delivery"—Dr. Traynor identifies three further alleged triggers: (4) receipt by the MMS Relay/Server of a message without a specified delivery time, which "immediately initiates the notification/retrieval process for a delivery attempt"; (5) for a message with a specified delivery time, "the expiration of a timer or event occurrence at the delivery time," which "initiates the notification/retrieval process for a delivery attempt"; and (6) "[w]hen the MMS Relay/Server receives a retrieval request from the recipient User Agent with a request for deferred delivery, the expiration of a timer or event occurrence at the delivery time (trigger) initiates a delivery attempt from the recipient User Agent and determines that a specified deferred-delivery period is met or event occurrence at the delivery time, triggering an attempted delivery." Ex. 1003, ¶601.

78. Notably, Dr. Traynor does not opine that alleged triggers (4) or (5) are sufficient to cause a message delivery attempt; he opines only that they "initiate[] the notification/retrieval process." *Id.* As I described in my overview of TS-23.140, that notification/retrieval process comprises informing the User Agent that a message is available, the User Agent electing whether to reject, defer, or retrieve the message, the User Agent issuing a retrieval request, and only then delivery by

the MMS Relay/Server. Ex. 1004, 28-29. Conditions (4) and (5) are therefore not sufficient for message delivery, and are not “message delivery triggers” under the plain meaning of that term. They are also inconsistent with Dr. Traynor’s own reasoning for why message receipt does not satisfy the negative claim limitation: under his framework, receiving a message without a specified delivery time would not “alone” be a trigger, and the expiration of a timer would not “alone” be a trigger, because in each case delivery will occur only if the notification/retrieval sequence thereafter completes. See Ex. 1003, ¶603 (reasoning that “message receipt by the memory store/MMBox alone is not a trigger”).

79. Alleged trigger (6) fails for a more basic reason: TS-23.140 does not disclose “a retrieval request from the recipient User Agent with a request for deferred delivery” that starts a timer whose expiry triggers delivery. In TS-23.140, when message retrieval is deferred, there is no retrieval request at all at the time of deferral. Instead, in response to the notification from the MMS Relay/Server, the MMS User Agent issues a notification response indicating that retrieval of the message has been deferred (as opposed to the message being immediately rejected or retrieved). Ex. 1004, 67 (“The recipient MMS User Agent may indicate in the MM1_notification.RES how it intends the MM to be handled, e.g., the immediate rejection of the MM.”); id., 67 (FIG. 8). Then, if and when the User Agent decides that it wishes to retrieve the deferred message, it separately issues a retrieval

request, which results in delivery of the message. *Id.*, 67 (FIG. 8). The only other disclosed handling of a deferred message is that the User Agent may instead request that the deferred message be deleted. *Id.*, 61 (FIG. 7A); *id.*, 29 (“For any MM for which the retrieval has been deferred, the MMS User Agent may request deletion on the MMS Relay/Server instead of MM retrieval.”).

80. Nowhere does TS-23.140 disclose that a retrieval request includes a request for deferred delivery, or that deferral initiates any timer at the MMS Relay/Server that results in message delivery upon expiration. For deferred messages, TS-23.140 consistently teaches that the deferred message must either be retrieved via a subsequent retrieval request, or eventually be deleted due to message expiration or a deletion request. And as with alleged triggers (4) and (5), Dr. Traynor’s own reasoning confirms that none of the conditions in alleged triggers (4)-(6) “alone” results in delivery; in each case, other conditions—including reception of the message in the first place and a subsequent valid retrieval request—must also be satisfied. *See* Ex. 1003, ¶603.

4. Dr. Traynor’s alleged trigger (7) is part of the reception of a valid retrieval request.

81. Under his third and broadest tier—“conditions imposed on the messages controlling whether a message will be delivered at all”—Dr. Traynor identifies one further alleged trigger: “(7) When the MMS Relay/Server receives a

retrieval request from the recipient User Agent with a ‘size restriction,’ a determination that the message conforms to that restriction (trigger) initiates a delivery attempt.” Ex. 1003, ¶601. In my opinion, this is not a separate message delivery trigger. Every message retrieval request must be determined to be valid or not: this includes ensuring that there is either no size restriction or that any provided size restriction is met, and ensuring that a message is properly identified in the request. See Ex. 1004, 29; id., 71 (“Message Reference” is a “Mandatory” element of the retrieval request). Determining that a retrieval request is valid is part of receiving and processing the retrieval request—the single event that TS-23.140 discloses as initiating delivery—not a distinct trigger.

82. Alleged trigger (7) also illustrates the double-counting inherent in Dr. Traynor’s approach. The retrieval requests of alleged triggers (1) and (2) are the same retrieval requests that would carry any size restriction referenced in alleged trigger (7). Under Dr. Traynor’s counting, the reception of a retrieval request is one trigger, and the determination that the size restriction within that same request is met is another trigger. And presumably, a determination that the same request identifies a valid Message Reference would be yet another. In this way, the single message delivery trigger that TS-23.140 actually discloses—reception of a valid message retrieval request—is merely being creatively counted to manufacture the claimed “plurality of message delivery triggers.” In my opinion, a POSITA would

not understand TS-23.140 to disclose a plurality of message delivery triggers under any of Dr. Traynor's three tiers.

83. And again, Dr. Traynor's own reasoning confirms that alleged trigger (7) "alone" does not result in delivery, because at least one additional condition—that the message is not time-expired—is required for message delivery. Thus, under Dr. Traynor's own logic, alleged trigger (7) can be no more a "message delivery trigger" than the message receipt that he implies is not one. *See* Ex. 1003, ¶603.

IX. OPINIONS REGARDING DR. TRAYNOR'S TS-23.140-BASED THEORIES FOR CLAIMS 3 AND 4

A. The "earliest desired time of delivery" is not the "expiration of a periodic timer."

84. Dr. Traynor notes that an originating MMS User Agent can set an "earliest desired time of delivery," in which case the MM is stored by the MMS Relay/Server until that time. Ex. 1003, ¶640. He then opines that "a POSITA would have understood or found it obvious that specifying the desired time of delivery triggers a periodic timer upon the receipt of a message that expires when the delivery time is reached," and that "[s]uch a timer would be periodic in that each time a message is received, the timer is triggered and continues for a particular period at the expiration of which, the message is delivered to the intended recipient." Ex. 1003, ¶641.

85. Even accepting Dr. Traynor’s characterization, reaching an “earliest desired time of delivery” is not the “expiration of a periodic timer.” As I explained in my claim construction discussion above, a “periodic timer” is a timer that recurs at regular intervals—not merely a timer that “continues for a particular period.” A one-shot timer that is armed when a message arrives and expires at that message’s earliest desired delivery time runs exactly once for that message, and it does not repeat or recur at a regular interval. Dr. Traynor identifies no disclosure in TS-23.140 that any “earliest desired time of delivery” timer would recur at regular intervals. Notably, his reasoning here stands in sharp contrast to his periodic-polling theory, where he specifically reasons that “the periodic polling would be done at periodic intervals (i.e., after a certain time period has elapsed/expired) that repeat regularly.” Ex. 1003, ¶643.

86. To the extent Dr. Traynor’s opinion is that this configuration would have been obvious even if not disclosed (see Ex. 1003, ¶641, opining that a POSITA “would have understood or found it obvious”), I disagree, and I note that no obviousness rationale is provided: Dr. Traynor identifies no motivation to modify TS-23.140’s disclosures so that a recurring, regularly-repeating timer governs the earliest desired time of delivery, and he does not explain what any such modification would even entail. In my opinion, a POSITA would have been

aware of no reason to implement an earliest-desired-delivery-time function using a timer that recurs at regular intervals.

87. Dr. Traynor’s theory also reflects a misunderstanding of how TS-23.140 operates. As I explained in my overview of TS-23.140, the only way a message is delivered to an MMS User Agent in TS-23.140 is that the MMS User Agent first issues a valid retrieval request. Ex. 1004, 29. The retrieval of MMs—“initiate MM delivery to the MMS User Agent”—is among the application layer functionalities that “[t]he MMS User Agent shall provide.” Ex. 1004, 19. A POSITA would understand this to be a mandatory aspect of the specification, given TS-23.140’s use of “shall.”

88. Accordingly, allowing the expiration of a “desired delivery time” timer to itself initiate delivery of the message to the MMS User Agent—as Dr. Traynor’s theory requires for the timer expiry to be a “message delivery trigger” under my understanding of the term’s scope—would be inconsistent with TS-23.140’s explicit teachings, and would remove the receiving MMS User Agent’s autonomy in initiating MM delivery, which is a mandatory part of TS-23.140’s architecture. A POSITA implementing TS-23.140 would not read the earliest-desired-time-of-delivery feature as overriding the notification-and-retrieval sequence that TS-23.140 mandates for delivery to the User Agent.

89. I understand that it might be argued that, when the “earliest desired time of delivery” is reached, the MM is “routed forward” to the intended MMS User Agent recipient. *See* Ex. 1004, 27 (“Upon reception of an MM from an originator MMS User Agent the originator MMS Relay/Server ... shall retain the MM until the earliest desired time of delivery, if the optional feature of earliest time of delivery is supported by the originator MMS Relay/Server. If this feature is not supported then the MM is immediately routed forward....”). That reading would be inconsistent with TS-23.140’s actual disclosure. This passage of TS-23.140 concerns routing/forwarding an MM between two different MMS Relay/Servers—the originator MMS Relay/Server and the recipient MMS Relay/Server. This is made clear by TS-23.140’s teaching later in the same section that “[a] special case is where the recipient MMS Relay/Server is also the originator MMS Relay/Server. In this case the MM does not have to be routed forward.” Ex. 1004, 28. In that special case, the MMS Relay/Server simply does not provide a message notification to the recipient MMS User Agent until the earliest desired time of delivery has passed; the notification-and-retrieval sequence then proceeds as usual.

90. In other words, “rout[ing]” a message “forward” in this passage refers to routing the message between MMS Relay/Servers (over reference point MM4), not to immediate delivery of the message from an MMS Relay/Server to an MMS

User Agent (over reference point MM1) when the earliest desired time of delivery is reached. *See* Ex. 1004, 91 (§8.4.1, “Routing Forward of a Multimedia Message,” describing MM4_forward.REQ between an “originator MMS Relay/Server” and a “recipient MMS Relay/Server”). By having the originating MMS Relay/Server “retain the MM until the earliest desired time of delivery, if the optional feature of earliest time of delivery is supported” (Ex. 1004, 27), the originating MMS Relay/Server guarantees that the recipient MMS Relay/Server—the server that will ultimately be responsible for delivering the MM to the recipient MMS User Agent—cannot deliver the message before the earliest desired time of delivery, because the recipient server does not yet have the message.

91. Accordingly, while an “earliest desired time of delivery” setting could potentially serve as a trigger for a first MMS Relay/Server to forward an MM to another MMS Relay/Server, neither TS-23.140 nor Dr. Traynor explains how it would be a trigger for the MMS Relay/Server to deliver a message to an MMS User Agent, as the claims require. *See* Ex. 1001, limitation [1d4] (“upon determining that one of the message delivery triggers has occurred for the given one of the wireless end-user devices, the logic further to supply one or more messages comprising the buffered content ... for delivery on the secure message link maintained between the transport services stack and a device link agent on the given one of the wireless end-user devices”). For all of these reasons, it is my

opinion that the earliest-desired-time-of-delivery theory does not establish that “one of the message delivery triggers is the expiration of a periodic timer.”

B. TS-23.140’s “periodic polling” does not satisfy claim 3.

92. Dr. Traynor’s second theory is that the “MMS User Agent uses periodic polling to identify new messages on an external server, and subsequently retrieves those messages,” which he contends is “another example [of] a message delivery trigger that is the expiration of a periodic timer.” Ex. 1003, ¶642; see id., ¶643. At his deposition, Dr. Traynor acknowledged that his declaration does not provide any opinion about how the alleged polling would be achieved, other than via a “message retrieval request.” Ex. 2010, 18:17-19:2. In my opinion, there is no teaching in TS-23.140 that the MMS User Agent engages in the alleged periodic polling, and that understanding would be inconsistent with TS-23.140’s teachings.

93. The passage on which Dr. Traynor relies is Section 8.3.3 of TS-23.140, which states: “For discovery of incoming messages from external messaging systems different mechanisms may be utilised, e.g.: ... periodic polling for messages on External Server, followed by retrieval by the MMS User Agent via the MMS Relay/Server.” Ex. 1004, 90-91.

94. This passage does not say that the MMS User Agent performs the periodic polling. It says that periodic polling is one mechanism for “discovery of incoming messages from external messaging systems,” and that such polling is

“followed by retrieval by the MMS User Agent via the MMS Relay/Server”—i.e., the User Agent’s retrieval is a separate, subsequent step. A POSITA would understand that it is the MMS Relay/Server’s role—not the MMS User Agent’s—to interact with external messaging systems (“External Servers”). TS-23.140 is explicit on this point: “Convergence functionality between External Servers and MMS User Agents is provided by the MMS Relay/Server which enables the integration of different server types across different networks.” Ex. 1004, 22.

95. The architecture of TS-23.140 confirms this division of roles. Figures 2 and 4 of TS-23.140 both show the MMS User Agent separated from the External Servers by the MMS Relay/Server, with no direct communication path between the MMS User Agent and any External Server. Ex. 1004, 17 (FIG. 2), 24 (FIG. 4). The MM1 reference point connects the MMS User Agent to the MMS Relay/Server; the MM3 reference point connects the MMS Relay/Server to External Servers. *Id.*

96. TS-23.140 also provides a specific example of how polling of an external messaging system works in practice: for interworking with a voice mailbox system, “the MMS Relay/Server may poll the voice mailbox via POP3 or IMAP4 for new messages received.” Ex. 1004, 157-158. Nowhere does this example—or any other teaching of TS-23.140—indicate that the MMS User Agent would periodically poll an External Server, as Dr. Traynor’s opinion entails.

97. At his deposition, Dr. Traynor was asked whether “TS 23.140 actually discloses that the MMS User Agent performs the periodic polling.” He did not answer in the affirmative; he instead stated that User Agent polling would be “the scalable way to do it,” while acknowledging that “[p]otentially, the MMS Relay/Server could poll other servers.” Ex. 2010, 33:17-34:10 When asked where his declaration addresses this scalability rationale, Dr. Traynor conceded, “I don’t believe I have any text on that exactly.” Ex. 2010, 35:9-21. In other words, the opinion that the MMS User Agent would perform the polling rests on a rationale that appears nowhere in his declaration—and, in my opinion, it is also technically wrong in the context of TS-23.140.

98. Specifically, Dr. Traynor’s assertion that the periodic polling—which he testified would take the form of “retrieval requests” (Ex. 2010, 18:17-19:2)—would be performed by the MMS User Agent makes no sense in the context of TS-23.140. TS-23.140 is clear that the purpose of the periodic polling is the “discovery of incoming messages from external messaging systems.” Ex. 1004, 90-91. But a retrieval request cannot be used to discover unknown messages: a retrieval request must include a “Message Reference” (a “Mandatory” information element, Ex. 1004, 71), and TS-23.140 teaches that “[t]his reference was previously delivered to the MMS User Agent” in a notification or other message sent by the MMS Relay/Server. Ex. 1004, 69. Without already being aware of an incoming

message—via a notification from the MMS Relay/Server—it would be impossible for an MMS User Agent to issue the retrieval requests that Dr. Traynor equates with polling. Accordingly, the alleged User-Agent “periodic polling” could not be used for “discovery of incoming messages from external messaging systems” as TS-23.140 describes.

99. My reading is also consistent with the stated purpose of Section 8.3 of TS-23.140, which “defines the interworking between MMS Relay/Servers and External Messaging Servers” over reference point MM3. Ex. 1004, 90. Section 8.3.3’s “periodic polling” is thus naturally understood as an MM3-side mechanism performed by the MMS Relay/Server. I additionally note that TS-23.140 defines an External Server as connected “via non-MMS-specific protocols” (Ex. 1004, 14), and Dr. Traynor provides no analysis that a message received from an External Server would meet the requirements the claims impose on the received network element messages—including that such messages be received “for delivery to respective ones of the software components identified in the messages” (Ex. 1001, limitation [1c1]), and that at least a first subset of the resulting secure Internet data messages “contain both a unique identifier for a corresponding one of the software agents and data to be consumed by that software component” (*id.*, limitation [1e3]).

100. Finally, even if the MMS User Agent did engage in periodic polling, the theory would still not satisfy the claims. If User-Agent polling is “one of a plurality of message delivery triggers,” the network server system must have “logic to determine when” that trigger “has occurred.” Ex. 1001, limitation [1d3]. From the MMS Relay/Server’s perspective, an alleged “poll” would arrive simply as a retrieval request, and Dr. Traynor identifies no logic in TS-23.140 by which the MMS Relay/Server could determine whether any given retrieval request resulted from periodic polling as opposed to any other cause. At his deposition, Dr. Traynor was unable to identify any such logic. Ex. 2010, 36:16-23 (“Q. Does TS 23.140 disclose any logic to determine whether or not a given message retrieval request was the result of periodic polling? THE WITNESS: ... I’m not aware off the top of my head.”).

C. The proposed combination of TS-23.140 and Adamczyk does not render claims 3 and 4 obvious.

101. Dr. Traynor alternatively opines that claim 3 would have been obvious over TS-23.140 in combination with Adamczyk. His theory is that, consistent with Adamczyk’s disclosure that notification servers can be “configured to send [] notification messages to the recipients on demand, at a specific future time, and/or on a periodic schedule” (Ex. 1003, ¶690, quoting Ex. 1021), “a POSITA would have been motivated to implement a periodic timer like

Adamczyk's to have the MMS Relay/Server issue notifications at pre-determined intervals [with] all messages received during that interval." Ex. 1003, ¶692. (Dr. Traynor confirmed at his deposition that the word "will" in this sentence of his declaration should be read as "with." Ex. 2010, 37:24-38:10.)

102. Dr. Traynor confirmed at his deposition that his theory is not that messages would be automatically delivered upon expiration of the periodic timer, but only that notification messages—informing the User Agent that messages are available for retrieval—would be sent according to a periodic interval:

Q. Okay. So is it your opinion that on expiration of the periodic timer, the MMS Relay/Server would issue a notification to the User Agent that a message is available and then, in response to that notification, the User Agent could request delivery of that message?

THE WITNESS: Again, on expiration of the timer at the MMS server, it -- a notification that messages were available would be sent. If the device on which the User Agent was running was available, you know, met a number of criteria and, you know, including things like had sufficient storage for the messages, it then might request to receive them.

...

Q. ... My question is, is your proposed combination with TS 23.140 changing the operation of TS 23.140 such that upon expiration of the periodic timer, the MMS Relay/Server would just transfer the application message to the User Agent without first waiting for a retrieval request?

A. I -- I don't believe that I'm suggesting that. I'm saying that the expiration of a timer triggers a series of additional events, each of which could potentially stop the delivery. So if a notification was sent to the User Agent and it didn't respond, that -- those messages would

not be delivered. So I'm just saying that there's a -- a trigger that's happening at the MMS server.

Ex. 2010, 41:19-42:24.

103. As I explained in my claim construction discussion, sending a notification to the MMS User Agent is not a "message delivery trigger" under my understanding of that term, because the notification is insufficient to trigger message delivery: as Dr. Traynor himself testified, after the notification is sent, "a series of additional events" must occur, "each of which could potentially stop the delivery." Ex. 2010, 42:17-24 Accordingly, even if the proposed combination of TS-23.140 and Adamczyk were made, the periodically timed notifications would not make the timer's expiration a "message delivery trigger," and the combination would not meet the requirements of claim 3.

104. In my opinion, a POSITA also would not have been motivated to make the proposed combination. Dr. Traynor's proposal is that the MMS Relay/Server would issue notifications at pre-determined intervals covering "all messages received during that interval." Ex. 1003, ¶692. That configuration conflicts with TS-23.140's earliest-desired-delivery-time feature: a notification schedule keyed only to a recurring interval (as Dr. Traynor proposes) would send notifications for messages whose earliest desired time of delivery has not yet arrived. For example, if the MMS Relay/Server were configured to transmit

notifications at the top of every hour for all messages received in the preceding 60 minutes, then for a message received at 1:30 PM with an earliest desired delivery time of 10:30 PM, the proposed combination would issue a notification at 2:00 PM—inconsistent with TS-23.140’s teaching that the message be withheld until the earliest desired time of delivery. Where the earliest-desired-time-of-delivery feature is supported, the modification would thus violate the feature’s purpose; Dr. Traynor does not address this conflict or explain how his proposed configuration would be reconciled with it.

105. For messages without a specified earliest desired time of delivery, the proposed modification would have the clear drawback of unnecessarily delaying notifications—messages that TS-23.140 would otherwise notify immediately upon reception (Ex. 1004, 28) would instead wait for the next periodic interval. Dr. Traynor identifies two alleged benefits that would supposedly justify this drawback: “additional message delivery triggers create a more configurable system that better suits a wider range of user needs in an efficient manner” (Ex. 1003, ¶692), and “periodic timers would arguably improve the battery life of the device operating the MMS User Agent” (Ex. 1003, ¶693). In my opinion, neither withstands scrutiny. As to the first, Dr. Traynor does not explain why a user would find it beneficial for notifications to be batched to periodic intervals, thereby either delaying their arrival (if no earliest delivery time is specified) or overriding the

earliest desired delivery time (if one is). And the resulting system would in a meaningful sense be less configurable, because it would defeat the ability to give effect to the earliest desired time of delivery, as discussed above.

106. As to the alleged battery-life benefit, Dr. Traynor’s reasoning rests on a mischaracterization of how TS-23.140 operates. His stated premises are that, absent the modification, the system would either “have to establish a connection between the MMS User Agent and MMS Relay/Server every time a message is received by the MMS Relay/Server,” or “have the MMS User Agent repeatedly ping the MMS Relay/Server to see if messages are pending.” Ex. 1003, ¶693. But TS-23.140 does not disclose the MMS User Agent “repeatedly ping[ing]” the MMS Relay/Server to check for pending messages—as I explained above, TS-23.140’s notification mechanism means the User Agent learns of new messages from the MMS Relay/Server, and the “periodic polling” disclosed in Section 8.3.3 is polling of External Servers on the MM3 interface, which a POSITA would understand to be performed by the MMS Relay/Server. There are, accordingly, no “repeated pull requests” in TS-23.140 for the proposed modification to eliminate.

107. Nor would a POSITA have accepted the premise that a new connection must be established for every message. TS-23.140’s MM1 transactions are carried over standard Internet protocols (e.g., an HTTP-based protocol over TCP/IP), and a POSITA would have understood that such connections can be

maintained efficiently rather than re-established for each message—an approach that Dr. Traynor himself embraces elsewhere in his declaration, where he opines that a POSITA would have been motivated to implement “keep-alive timers” and keep-alive messages so that connections between the User Agent and Relay/Server are maintained “for as long as the MMS User Agent indicates it is available.” Ex. 1003, ¶¶697-698. Houghton—another reference on which Dr. Traynor relies in this proceeding—confirms this was the understood engineering practice, explaining that “[t]he cost for a user’s mobile terminal to maintain a connection to the push server and receive such messages over mobile Internet connections is minimal,” and that messages sent “through a previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS arrive quite rapidly, more rapidly than if the connection w[ere] established only when needed,” because the “associated protocol overhead ... occurs before the actual need for message transmission.” Ex. 1007, 12, 20.

108. Given a maintained connection, batching notifications to pre-set intervals does not meaningfully reduce the amount of data transmitted: the same notifications are transmitted in either scenario, merely at different times. Consistent with this, Dr. Traynor’s declaration asserts only that periodic timers “would arguably improve the battery life” (Ex. 1003, ¶693 (emphasis added))—and the only evidence he cites concerns “setting intervals for call connections

rather than persistent connections” (Ex. 1003, ¶693, citing Ex. 1040, [0074]). A POSITA would not view that evidence as relevant here: Ex. 1040 addresses an “audio voice communications channel” that is separate from “the additional TCP data communications channel” (Ex. 1040, [0039]), and there is no evidence that voice call connections could be maintained with anything approaching the efficiency of an idle TCP connection with occasional keep-alive messages. In my opinion, neither Dr. Traynor’s declaration nor the materials he cites set forth any plausible technical basis for the claimed battery-life benefit, and a POSITA would not have accepted a certain drawback (delayed message notifications) in exchange for a speculative, “arguabl[e]” benefit of unspecified magnitude.

109. Finally, in my opinion a POSITA would not have had a reasonable expectation of success in making the combination as proposed. Dr. Traynor’s proposed configuration—notifications issued at pre-determined intervals covering all messages received during the interval (Ex. 1003, ¶¶692, 694)—is inconsistent with TS-23.140’s handling of the earliest desired time of delivery where that feature is supported. *See* Ex. 1004, 27 (“Upon reception of an MM... the originator MMS Relay/Server... **shall** retain the MM until the earliest desired time of delivery.... *If this feature is not supported then the MM is immediately routed*

forward.”).¹ Neither Dr. Traynor nor the references articulate how the proposal would be reconciled with that operation. For all of these reasons, it is my opinion that the TS-23.140-Adamczyk theory fails as to claim 3; because claim 4 depends from claim 3, the theory fails as to claim 4 as well. I understand that Dr. Traynor’s claim 4 opinions based on Herzog (with or without Adamczyk) rely on TS-23.140’s periodic polling or Adamczyk’s periodic schedule to supply claim 3’s “periodic timer” (Ex. 1003, ¶¶695-696); those opinions accordingly fail for the same reasons discussed above, as Herzog is not relied upon to supply the “expiration of a periodic timer.”

X. OPINIONS REGARDING DR. TRAYNOR’S TS-23.140-BASED THEORY FOR CLAIM 17

A. Dr. Traynor fails to show it would be obvious to combine TS-23.140 with Fok in his proposed manner.

110. Neither Fok nor TS-23.140 discloses a two-handshake process to deliver a single message; in Fok, applications communicate directly with one another. This, however, would not satisfy the claim requirements, which is presumably why Dr. Traynor does not propose modifying the TS-23.140 device with what is actually disclosed in Fok.

¹ As I discuss above, “[a] special case is where the recipient MMS Relay/Server is also the originator MMS Relay/Server. In this case the MM does not have to be routed forward.” Ex. 1004, 28. In that special case, the MMS Relay/Server simply does not provide a message notification to the recipient MMS User Agent until the earliest desired time of delivery has passed; the notification-and-retrieval sequence then proceeds as usual.

111. In my opinion, Dr. Traynor’s deviation from Fok’s single-handshake mechanism to implement a two-handshake procedure with the MMS User Agent serving as a middleman would not have been obvious. I disagree with Dr. Traynor’s stated rationales in this regard. For example, Dr. Traynor alleges that his proposed deviation from Fok’s teachings would (1) “effectively provide [sic] MMS-compliant applications to interact without needing to develop, for instance, custom APIs (‘Application Programming Interfaces’) for alternative forms of direct communication,” and (2) “ensure the applications were trusted before sharing information; the MMS User Agent would provide the interface and relevant authorizations.” Ex. 1003, ¶757.

112. Regarding the “custom APIs” aspect of Dr. Traynor’s opinion, Dr. Traynor presents no evidence that in a device implementing TS-23.140, *direct* communication between applications (rather than using the MMS User Agent as a middleman as Dr. Traynor proposes) would require “custom APIs” above and beyond what would be required for Dr. Traynor’s proposal (which deviates from the teachings of Fok). But there is no reason that the *same* API already used to transmit data to the MMS User Agent could not be used to transmit data to other applications directly, and Dr. Traynor identifies none. For example, TS-23.140 does not specify any protocol for transferring data between the MMS User Agent and individual applications, such that the API used for such a transmission need

not be exclusive to MMS transmissions. *See* Ex. 1004, 54 (“Details of these applications or how an MMS User Agent or an MMS VAS Application would interface with them are outside the scope of this specification.”). Thus, to the extent the applications used APIs for communication at all, there is no reason that the *same* API used for communication between the MMS User Agent and other applications could not be used for communication with individual applications residing on a device, such that no new “custom APIs” would need to be developed “for alternative forms of direct communication” as Dr. Traynor contends. And certainly, there is no evidence that a POSITA would have recognized development of custom APIs as a problem that would require deviation from Fok’s teaching of applications directly communicating with multiple different applications within a device. *See* Ex. 1038, [0047] (noting that a given application “can maintain multiple secure communication sessions with multiple other applications... concurrently”).

113. Regarding Dr. Traynor’s allegation that his proposed deviation from Fok’s teachings would “ensure the applications were trusted before sharing information” because “the MMS User Agent would provide the interface and relevant authorizations” (Ex. 1003, ¶757), Dr. Traynor does not explain how this would solve any problem in the direct application-to-application handshake mechanism that Fok already discloses. For example, as noted above, applications

would *already* need a communication interface for communicating with the MMS User Agent, and Fok already teaches that a single application can communicate with multiple other applications, such that there is no reason why using Fok's handshake teachings *without* inserting the MMS User Agent as a middleman would have required a separate interface. And individual applications would *already* need to be configured to perform authorization functionality under Dr. Traynor's proposal, in order to engage with Fok's handshake mechanism (even if transmission was limited only to the MMS User Agent as Dr. Traynor proposes). Dr. Traynor is thus incorrect that the MMS User Agent would perform all relevant authorizations under his proposed combination—the individual applications would *also* need to be configured to provide authorization functionality, regardless of whether the MMS User Agent was used as a middleman component as Dr. Traynor proposes. *See, e.g.*, Ex. 1038, [0006] (noting that a “recipient application... reference[s] the unique identifier with a list of certified primary applications to verify the primary application”); *id.*, [0052] (discussing applications checking a unique ID to determine whether a primary application is “an un-authorized application”).

114. Nor is the use of the MMS User Agent as a middleman necessary to “ensure the applications were trusted before sharing information” as Dr. Traynor implies. As Fok discloses, typically, “an operating system (106) can require certain

standards be met for applications designed to execute on the operating system (106),” including “incorporat[ing] a list of unique identifiers of trusted primary applications” such that a “recipient application... could then verify that a received handshake request is sent by an application authorized to make the request” and the recipient application “could then trust and engage in communication with such application.” Ex. 1038, [0048]. In other words, Fok already teaches techniques to ensure that applications are trusted before sharing information, and the introduction of an unnecessary MMS User Agent middleman in the transmission process thus does not solve a problem in Fok.

115. Furthermore, I am aware of no evidence that implementing a handshake technique as described in Fok to require *two* handshakes for a single transmission between two applications (as opposed to one handshake, as proposed by Fok itself) was conventional. Nor do I believe this modification would have been obvious. For example, Dr. Traynor presents no evidence that a POSITA would have found Fok’s technique of direct communication between applications residing on a device to be deficient. Nor do I believe that a POSITA would have viewed Dr. Traynor’s proposed deviation from Fok’s teachings as an improvement over Fok’s teachings, even if a POSITA was for some reason looking to improve upon Fok’s teachings. A transmitting application must know which application it intends to receive the message (because it would need to include an application

identifier pursuant to the MMS protocol under Dr. Traynor's proposal), such that it would be able to directly address the recipient application using Fok's handshake technique as disclosed, without using the MMS User Agent as an unnecessary middleman with an additional handshake procedure.

116. I also disagree with other of Dr. Traynor's statements that could be viewed as alleged motivations to deviate from Fok's disclosures. For example, Dr. Traynor alleges his deviation from Fok's teachings would be advantageous because "each application would not need to be configured with a brand-new 'handshake' mechanism, but would instead utilize its existing protocols for MMS transmissions." Ex. 1003, ¶758. What Dr. Traynor's opinion ignores is that an application's "existing protocols for MMS transmissions" do not include handshake procedures at all. *See generally* Ex. 1004 (no discussion of handshakes at all, much less for communicating with the MMS User Agent or other applications on a device). Thus, a POSITA attempting to implement *any* usage of Fok's handshake techniques for intra-device communication would need to configure each participating application with "a brand-new handshake" mechanism (i.e., Fok's handshake mechanism that Dr. Traynor himself proposes would be used). And Dr. Traynor presents no reason why having applications communicate with each other directly (as opposed to communicating with the MMS User Agent) would require a *distinct* handshake mechanism. It would not, as Fok makes clear

that its handshake mechanism allows any participating application “can maintain multiple secure communication sessions with multiple other applications... concurrently” (Ex. 1038, [0047]), and a POSITA would have understood that the *same* handshake mechanism would be used for each of these concurrent sessions (as evidenced by Fok containing no disclosure that multiple distinct handshake protocols would be required for multiple concurrent sessions).

117. Furthermore, Dr. Traynor’s proposal would have the obvious drawback of doubling the number of handshakes and communications required for communication between two applications. For instance, under Fok’s disclosure, under Dr. Traynor’s first example where a “messaging application could direct a video-playback application to download a specific codec,” it would require the messaging application to initiate a handshake with the MMS User Agent and further require the MMS User Agent to subsequently initiate a handshake with the video-playback application in order to secure the message delivery. There is absolutely no evidence that a POSITA would have been motivated to implement this solution, and in my opinion a POSITA would have avoided doing so because it would require the number of communication sessions and handshakes to be doubled, increasing transmission time, processing power, and battery drain. *See* Ex. 1007, 20 (noting that “protocol overhead such as... handshake” should ideally “occur[] before the actual need for message transmission” because handshaking at

the time of transmission would “delay[] the transmission,” such that a POSITA would have avoided unnecessary handshakes because they lead to delay).

118. Another disadvantage to Dr. Traynor’s proposal (relative to just implementing Fok’s teachings within TS-23.140 without modifying them to involve the MMS User Agent as an intermediary) is that, as Dr. Traynor acknowledges, this would limit secure interprocess communication to “MMS-compliant applications.” *See* Ex. 1003, ¶757. To the extent a POSITA would have recognized secure interprocess communication within a device implementing TS-23.140 as an advantage in the first place, there is no reason they would not design the solution to allow for secure interprocess communication between all applications (not merely MMS-compliant applications). For example, a POSITA would recognize, to use Dr. Traynor’s example, that it would be common for an application such as a “[photo]-editing application” (Ex. 1003, ¶757) to not be MMS-compliant, but still benefit from communication with other processes within the end-user device. Dr. Traynor’s proposal to use the MMS User Agent as a middleman would not allow such an application to communicate with other applications such as a messaging application, but implementing Fok’s techniques to allow for direct communication between photo-editing application and messaging application **would** allow for the communication to occur, and would further accomplish that result with less signaling overhead.

B. Dr. Traynor does not establish that his proposal, even if obvious, satisfies the claim requirements.

119. Claim 17 requires that the device link agent “caus[es] messages to be securely delivered to a software process by initiating delivery of each such message on the secure interprocess communication service.” Thus, messages must be delivered on an “interprocess communication service,” and that service itself must be “secure.”

120. Dr. Traynor does not identify *any* alleged “service” in either TS-23.140 or Fok that he contends would constitute a claimed “interprocess communication service.” Ex. 1003, ¶¶754–760 (not using the term “service” once except in restating the claim language, and not identifying anything in Fok or TS-23.140 that allegedly constitutes the claimed “service”). As best I can understand, Dr. Traynor’s contention is that secure delivery to a software process inherently involves a “secure interprocess communication service.” Ex. 1003, ¶754 (discussing “a device link agent... on a given device causing messages to be securely delivered to a software process” as allegedly satisfying the “secure interprocess communication service” requirement). But as discussed above in Section VI.C regarding claim construction, interprocess communication need not involve an interprocess communication *service*, just as two individuals

communicating with one another need not involve a communication service such as telephone or mail service.

121. Furthermore, Dr. Traynor would *additionally* need to show that any “interprocess communication service” identified is itself “secure.” But Dr. Traynor does not explain how any *communication service* used for communication is itself secure. Indeed, Fok itself seems to assume that to the extent any service is used for communication, that service is not itself secure, such that applications must identify their *own* form of security through Fok’s handshake process.

122. Thus, Fok discloses secure communication via a handshake process. But this handshake process is not a “communication service” provided to the applications—instead, it is a security mechanism implemented by each individual application and coordinated between applications. For example, consider an unsecured public Wi-Fi connection over a Starbucks Wi-Fi hotspot. The Wi-Fi hotspot provides a “communication service” between a user’s laptop and the Internet, but that “communication service” is not secure because it involves a public hotspot that does not provide a secure connection. Nevertheless, if messages transmitted over that unsecured communication service are secured via encryption mechanisms, secure delivery of messages may still be possible over the unsecured communication service. The fact that it is possible to securely transmit messages over an unsecured communication service does not transform an unsecured Wi-Fi

connection into a “secure communication service”; rather, it emphasizes that any security of transmission comes not from the communication service itself but from security mechanisms (such as encryption protocols) handled between the sending and receiving entities.

123. In the same manner, a POSITA would understand that even if Dr. Traynor *had* identified some “communication service” in Fok or TS-23.140 that provides communications between processes, any security of his proposal comes from handshaking mechanisms implemented by the applications themselves rather than security that is derived from whatever “communication service” is used to deliver messages. In my opinion, this is insufficient to satisfy claim 17, which requires that the communication service itself be secure (rather than the processes communicating over the communication service provide additional security mechanisms that are not themselves part of the provided interprocess communication service).

124. In short, Dr. Traynor neither identifies an “interprocess communication service” nor explains how any such service would itself be “secure” as the claims require.

XI. OPINIONS REGARDING DR. TRAYNOR'S HOUGHTON/MUNSON-BASED THEORIES

A. Dr. Traynor does not show that, for at least some received network element messages, receipt is not a message delivery trigger.

125. All challenged claims require that “for at least some of the received network element messages, the receipt of such a message by the message buffer system is not a message delivery trigger.” Ex. 1001, claim 1[d3]. For the Houghton/Munson combination, Dr. Traynor’s opinions on this limitation are set forth at paragraphs 841-853 of his declaration. There, he opines that Houghton’s “push message from the push server 401 may be triggered by any trigger event, local or remote, defined at the server,” and that the trigger “may, among others, include an alarm, notification, or measurement result received to the push server 401 from another device or system.” Ex. 1003, ¶842 (quoting Ex. 1007, 21). From this, he concludes that “the triggers noted above—alarm, notification, or measurement result—are message delivery triggers and do not constitute a trigger that is based on receipt of the message by the push server.” Ex. 1003, ¶844.

126. In my opinion, this analysis does not address—much less establish—the actual claim requirement. The claims do not merely require the existence of some triggers that are not message receipt. Instead, the claims require that, *for at least some of the received network element messages, the receipt of such a message is not a message delivery trigger*. Showing that the combined system also

has ***additional*** triggers (alarms, notifications, measurement results) says nothing about whether the receipt of any received network element message fails to qualify as a message delivery trigger. Dr. Traynor nowhere identifies any received network element message in the Houghton/Munson combination—i.e., a message received from a network element “for delivery to respective ones of the software components identified in the messages” (Ex. 1001, limitation [1c1]), the content of which is buffered because “delivery is requested to any of the wireless end-user devices” (id., limitation [1d2])—whose receipt is not a message delivery trigger. I also note that the “alarm, notification, or measurement result” triggers he identifies are themselves things “received to the push server 401 from another device or system” (Ex. 1007, 21)—that is, they arrive at the push server as received communications, which further blurs any distinction his analysis attempts to draw between triggers and received messages.

127. Moreover, in my opinion, Dr. Traynor could not have made the required showing consistent with his own interpretation of “message delivery trigger.” As discussed above, under his framework a message delivery trigger includes any “condition[] imposed on the messages controlling whether a message will be delivered at all” (Ex. 1003, ¶601), and any “condition that must occur before [the] message is delivered” (Ex. 2012, 34). In Houghton’s push-messaging system—as in any messaging system—a message that has not been received by the

push server cannot be pushed to a mobile terminal; receipt of each message at the push server is a condition that must occur before that message is delivered, and one of the first “traffic light[s]” on the way to delivery. Under Dr. Traynor’s own interpretation, then, the receipt of every received network element message in the Houghton/Munson combination is a message delivery trigger, and the negative limitation is not satisfied. In my opinion, his analysis of this limitation for Houghton/Munson (Ex. 1003, ¶¶841-853) simply restates the limitation in altered form—triggers “in which” or “that are not” message receipt—and never engages with the claim language as written.

B. A POSITA would not have combined Houghton and Munson as Dr. Traynor proposes.

128. Dr. Traynor opines that a POSITA would have found it obvious “to incorporate Munson’s store-and-forward functionality into Houghton’s push messaging system” (Ex. 1003, ¶761), motivated by a desire “to improve flexibility and reliability of the system” (*id.*, ¶762), and that the combination “yields predictable results (push messaging systems that can store and forward messages and subsequently deliver them ...).” *Id.*, ¶765. As I explain below, each of the proffered reasons to combine rests on a mischaracterization of Houghton or on generic benefits untethered to Houghton’s stated design goals—and the “predictable results” formulation merely restates the proposed end point of the

combination (a push-messaging system that stores and forwards messages) as if it were the reason a POSITA would pursue it.

129. What Dr. Traynor's analysis does not mention is that, while Houghton "contemplates" store-and-forward messaging systems (Ex. 1003, ¶763, citing Ex. 1007, 3, 7), Houghton is critical of such systems. Houghton's discussion of store-and-forward messaging at page 3 states:

SMS and MMS (Multimedia Messaging Service) are the most common forms of "push" messaging in the mobile industry. These store and forward messaging systems run by mobile telecommunications operators push all or part of the data to the mobile terminal over closed telecommunications signaling networks such as SS7 (Signaling System Seven). These signaling networks are closed; they are not part of the Internet. Each operator controls and encrypts this traffic, charging high prices for their use and limiting their use and usefulness.

Ex. 1007, 3.

130. Houghton then expressly contrasts its own invention with the store-and-forward delivery model:

IP push messages utilizing the principles of the invention are delivered rapidly, bypassing the public and often crowded SMS, MMS, WAP Push and email store-and-forward mechanism within an operator and between operators. This guarantee of very rapid event push provides usability benefits to users which may be then designed into the mobile applications. For example the ability to expect prompt delivery allows modification to interactive mobile application system design.

Ex. 1007, 13.

131. Houghton thus describes store-and-forward messaging systems as having limited “use and usefulness,” describes their delivery mechanisms as “often crowded,” and touts as the defining advantage of its own invention that its IP push messages are “delivered rapidly, bypassing” those very mechanisms—a “guarantee of very rapid event push” that provides “usability benefits to users which may be then designed into the mobile applications.” Ex. 1007, 3, 13. In my opinion, Houghton does not merely express a preference for a different transport network; it criticizes and discourages the store-and-forward delivery model itself, because that model defeats the prompt, deterministic delivery that Houghton identifies as its central benefit. A POSITA reading Houghton would understand that Houghton’s entire design premise is that push messages should not sit in an intermediary store awaiting later delivery conditions.

132. To be clear, my opinion is not that a base reference must be defective before a POSITA would consult other art, nor that Houghton forbids any storage of messages—Houghton itself retains “lazy or low priority messages” for later, cost-efficient transmission when a higher-bandwidth connection becomes available. Ex. 1007, 33-34 (claim 24). The point is a technical one about the proposed combination: Dr. Traynor proposes to interpose Munson’s store-and-forward delivery model—holding received messages at the push server and delivering them only upon later conditions—into the delivery path of the time-critical push

messaging that is Houghton's reason for being. In Houghton, a "trigger event" is what causes the push server to generate and send a push message in the first place; the message delivery follows the trigger immediately, except where efficient network transmission is required e.g., for lazy or low priority messages that require efficient transmission. Ex. 1007, 21, 33 (claim 24). Munson's model inverts that sequence: messages that have already been received, by default, sit in storage awaiting some later delivery condition. Grafting that model onto Houghton would delay exactly the messages Houghton promises will be "delivered rapidly." In my opinion, a POSITA designing Houghton's rapid-push system would not have made a modification that runs counter to Houghton's guarantee of rapid delivery based on the reasons Dr. Traynor identifies, which I address next.

133. Dr. Traynor's first specific motivation is that store-and-forward functionality would "avoid losing or discarding of messages (e.g., when the mobile terminal is outside network range or connectivity and the message is therefore undeliverable)." Ex. 1003, ¶763. In my opinion, the premise—that in Houghton, messages would otherwise be lost when a terminal is unreachable—mischaracterizes Houghton. Houghton's system maintains a "permanent IP connection" to each mobile terminal, and the mobile terminal "may also automatically re-establish the connection when the connection gets broken and adjust its connection settings based on past experience with failure to connect and

such broken connections.” Ex. 1007, 11. And where a message should not be sent immediately, Houghton retains it: Houghton’s push server “uses the connection availability information as a trigger event for sending lazy or low priority messages including application updates and application data updates routed through the higher bandwidth capacity device to the wireless terminal”—that is, it holds such messages until a suitable connection is available, rather than losing them. Ex. 1007, 33-34 (claim 24).

134. Dr. Traynor does not engage with these disclosures, does not identify any scenario in which Houghton’s own architecture actually loses a message, and offers no evidence that message loss was a recognized problem in Houghton’s architecture. Dr. Traynor posits that a recipient mobile terminal “may not have network connectivity at the time a message originator initiates delivery” (Ex. 1003, ¶763), but he does not address Houghton’s disclosures that the connection is automatically re-established when it is broken and that messages requiring efficient transmission are held until a suitable connection becomes available. Ex. 1007, 11, 33-34 (claim 24). In my opinion, a motivation premised on solving a problem that has not been shown to exist in the base reference reflects hindsight rather than the perspective of a POSITA at the time.

135. Dr. Traynor’s second specific motivation is that “by providing computer memory to store messages mid-transmission, both the originator and

recipient would be able to better control the conditions of message delivery.” Ex. 1003, ¶764. Dr. Traynor’s accompanying discussion does not say which delivery “conditions” a POSITA would have wanted to “control,” for which messages, or why any such control would have been viewed as an improvement in a system whose declared value is that push messages arrive as rapidly as possible—Houghton touts its users’ ability to “expect prompt delivery,” not to defer it. Ex. 1007, 13. Neither Dr. Traynor nor Munson identifies a single specific option to control delivery times or conditions that a POSITA would have viewed as beneficial in the context of Houghton’s system. In my opinion, unexplained “flexibility” or “reliability” of this kind would not have motivated a POSITA to restructure the delivery path of Houghton’s rapid-push system, and Dr. Traynor’s contrary opinion does not reflect the technical reality of the references.

C. Claims 3 and 4: Houghton-Munson (with or without TS-23.140).

136. Claim 3 requires that “one of the message delivery triggers is the expiration of a periodic timer.” For the Houghton/Munson combination, Dr. Traynor points first to Houghton’s disclosure of “a periodic message” sent so that devices in the communication path “do not time expire the connection,” which he says would “obviously be triggered sent at ‘the expiration of a periodic timer’” (sic). Ex. 1003, ¶882. But these periodic messages are keep-alive messages—

Houghton describes “periodically testing the connection with a suitable keepalive message, a ‘ping’, which resets timers in intervening operator stateful firewalls and/or NAT ... firewalls.” Ex. 1007, 11 (quoted at Ex. 1003, ¶882). Dr. Traynor does contend that Houghton’s periodic message is a message delivery trigger. Ex. 1003, ¶929 (“Houghton discloses using a ‘periodic message’ as a message-delivery trigger to keep a connection alive”). Neither he nor Houghton, however, explains how the expiration of a keep-alive interval would cause the delivery of any buffered message content. At his deposition, Dr. Traynor’s testimony on this point was equivocal. Asked whether the periodic message described at page 26 of Houghton constitutes a message delivery trigger, he answered that “the timer associated with the periodic message could be a trigger,” that “[t]he message itself might trigger—might serve as a trigger to the actual device,” and that “there are multiple ways in which it could be a trigger.” Ex. 2010, 55:24-56:6. Asked where Houghton discloses that the expiration of the periodic timer triggers the delivery of any message to the user device, he identified no such disclosure, describing instead “one possible embodiment that may send a periodic message” that “could be embodied multiple ways.” Ex. 2010, 56:7-15. And asked whether, upon expiration of the timer, application messages intended for a particular device would be sent to that device, he answered only “[p]ossibly.” Ex. 2010, 57:4-8. In my opinion, Houghton’s periodic messages are not message delivery triggers, because they only

maintain the connection and do not cause delivery of any buffered message content. Nor are they conditions that “must occur” before message delivery even under Dr. Traynor’s own claim interpretation (Ex. 2012, ¶55; *id.*, 34), because message delivery would remain possible without them—Houghton’s system is configured to “automatically re-establish the connection when the connection gets broken.” Ex. 1007, 11. Dr. Traynor adds that Houghton’s “periodic message timings” can be “measured in seconds, minutes or hours” and can be “adjusted based on the success or failure of earlier messages.” Ex. 1003, ¶883. That does not change my opinion: the interval at which a keep-alive message is sent says nothing about whether the expiration of that interval causes delivery of any buffered message content.

137. For the claimed “expiration of a periodic timer” as a message delivery trigger, Dr. Traynor instead relies on Munson. Specifically, Dr. Traynor opines that Munson discloses that “content can be pushed according to a schedule of time or event,” provides the example of a “group push” performed “during off-peak hours to take advantage of lower rates” (Ex. 1003, ¶884), and concludes that a POSITA “would have understood or found it obvious that ‘schedule[d] pushes during off-peak hours for maintenance would be triggered by the expiration of a regularly-repeating timer’” (*id.*, ¶885). For this theory Dr. Traynor quotes only paragraph [0044] of Munson, the entirety of which reads:

In one embodiment, a content can be pushed according to a schedule of time or event. For example, when used with a group push, an application service provider can use pushes (i.e., pushing contents) for a system maintenance purpose during off-peak hours to take advantage of lower rates.

Ex. 1017, [0044].

138. In my opinion, there are several independent problems with this theory. First, as discussed in the preceding section, the baseline Houghton/Munson combination would not have been obvious. That conclusion applies with additional force here, because the only example Munson gives of pushing “according to a schedule of time or event” is “when used with a group push” (Ex. 1017, [0044])—a functionality that Houghton does not disclose, and that Dr. Traynor does not contend it would have been obvious to add to Houghton.

139. Second, Munson does not disclose the use of a periodic timer to trigger message delivery. Munson discloses that an application service provider can decide to push content “according to a schedule of time or event.” Ex. 1017, [0044]. A schedule of time or event is not a periodic timer. As Amazon Technologies, Inc.’s own patent recognizes, an action can be performed “according to a published, but not necessarily periodic, schedule.” Ex. 2014, claim 3. A POSITA would likewise understand that pushing content “according to a schedule of time or event” does not imply—much less require—a regularly-repeating timer. For example, an application service provider might decide, with a holiday

approaching, to push holiday-related content; that push coincides with a schedule or event (the approaching holiday), not with the expiration of any timer. Likewise, pushing system maintenance updates “during off-peak hours” describes when a push, if needed, should occur (e.g., at some point after 9:00 PM on the day an update is ready); it would commonly be initiated by an employee of the application service provider when an update exists, and it neither requires nor implies a timer that expires at regular, recurring intervals. Off-peak maintenance pushes occur when there is maintenance content to push, and such maintenance may be irregular and infrequent rather than on every recurrence of a fixed period. And off-peak maintenance pushes need not use a timer at all.

140. Dr. Traynor’s contrary opinion—that a POSITA “would have understood or found it obvious” that Munson’s off-peak group pushes “would be triggered by the expiration of a regularly-repeating timer” (Ex. 1003, ¶885)—is stated without supporting evidence or technical reasoning, and I disagree with it for the reasons above. And Dr. Traynor identifies no *motivation* to modify Munson’s schedule- or event-based pushing to use a periodic timer. This is fatal to Dr. Traynor’s articulation of his theory, because Munson’s alleged “use of periodic timers for message delivery” is Dr. Traynor’s stated basis for the combination: “Given Houghton’s disclosures of delivery triggers and Munson’s use of periodic timers for message delivery, in my opinion, a POSITA would have been motivated

and would have had a reasonable expectation of success in implementing Munson's teaching of using periodic timers....” Ex. 1003, ¶885. If Munson does not use periodic timers for message delivery, the premise underlying Dr. Traynor's motivation disappears. Dr. Traynor also asserts that a POSITA would have been motivated to use periodic timers as a message delivery trigger “to enable more efficient use of network resources and avoid repeated message requests that can impact performance of network communications.” Ex. 1003, ¶885. Dr. Traynor does not explain why that rationale would apply to Houghton, and in my opinion it does not fit Houghton's system. Houghton does not disclose repeated message requests from the push client that a periodic delivery timer would eliminate: Houghton maintains a permanent IP connection over which the push server sends messages when trigger events occur, and the only periodic messages Houghton discloses on that connection are keep-alive messages, which a periodic delivery timer would not displace. Ex. 1007, 11, 21, 26.

141. Third, even if Munson did disclose periodic-timer-triggered pushes and the references were combined, the claims would still not be satisfied. Claim 1 requires the network server system to include “logic to determine when one of a plurality of message delivery triggers for a given one of the wireless end-user devices has occurred,” and claim 3 specifies that “one of the [plurality of] message

delivery triggers is the expiration of a periodic timer.” Accordingly, the server must have logic to determine when the periodic timer has expired.

142. In Munson’s example, it is the *application service provider* (an application server) that decides to push content according to its schedule. Ex. 1017, [0044]. Dr. Traynor identifies nothing in the combined Houghton/Munson *push server* that would have logic to determine when a message transmission was requested pursuant to a periodic timer: even if an application server-initiated content pushes according to its own periodic timer, the push server receiving those pushes would have no logic determining when such a periodically timed event took place. Rather, the push server would simply receive a message like any other, with no knowledge as to whether a periodic timer (or even any sort of timer) were used to trigger the push.

143. I understand Dr. Traynor also presents claim 1 under a further combination adding TS-23.140 (Ground 2B), and opines that the same combination would teach the remaining challenged claims for the reasons he gives in Grounds 2A and 2C-2G. Ex. 1003, ¶923. His TS-23.140-based additions concern application registration and application identifiers (id., ¶¶919-922); they are not directed to, and do not cure, any of the deficiencies I have identified with respect to claim 1 or claim 3. Because claim 4 depends from claim 3, my opinions apply equally to claim 4.

D. Claims 3 and 4: Houghton-Munson-Adamczyk.

144. Dr. Traynor alternatively opines that claim 3 would have been obvious over Houghton-Munson in further combination with Adamczyk, reasoning that “a POSITA would have been motivated to provide users the option to schedule the delivery of messages—including on a ‘periodic schedule,’ as taught by Adamczyk, sufficient to control when messages would be pushed to their devices.” Ex. 1003, ¶925. As an initial matter, I note that Dr. Traynor’s Ground 2C analysis identifies no benefit that a POSITA would have expected from the proposed modification. Unlike his TS-23.140-Adamczyk analysis, his Ground 2C analysis does not assert that the modification would yield a “more configurable system,” and does not assert any battery-life benefit. *Compare* Ex. 1003, ¶¶924-927 *with id.*, ¶¶692-693. In my opinion, the observation that Adamczyk describes a “periodic schedule” as one of several notification options does not supply a reason why a POSITA designing Houghton’s rapid-push system would have adopted one. And to the extent the configurability and battery-life rationales that Dr. Traynor offered for the TS-23.140-Adamczyk combination are understood to carry over to this ground, the motivation and reasonable-expectation-of-success opinions fail for generally the same reasons I discussed above with respect to TS-23.140 and Adamczyk—including that the alleged configurability benefit is unexplained and the alleged battery-life benefit rests on mischaracterizations of the underlying

systems and on evidence about voice-call connections (Ex. 1040) that a POSITA would not consider relevant to persistent data connections.

145. A POSITA also would not have recognized any benefit in modifying Houghton in the manner proposed. Houghton discloses that, with the exception of data that should be transmitted “in the most cost efficient manner” (e.g., “lazy application code update[s] and lazy application data update[s]” (Ex. 1007, 24)), messages in its system are to be transmitted as rapidly as possible, to serve the need for “real time information, interactivity and rich media content.” Ex. 1007, 10-11 (“Given the shortcomings in the prior art, there is a need for a method whereby a computer on the Internet can rapidly and at reasonable cost be permitted to at any time initiate control of one or more applications on an individual’s mobile terminal to deliver through that mobile application real time information, interactivity and rich media content....”). Transmitting messages upon the expiration of a periodic timer is antithetical to that goal: it would delay content that Houghton repeatedly describes as time-sensitive. *See, e.g.*, Ex. 1007, 10 (describing the unsuitability of “low speed” solutions for the contemplated messaging).

146. The sole exception Houghton makes to time-critical delivery—non-time-critical data such as lazy application code and data updates—does not help the theory either, because for that data Houghton already teaches a solution better

suited to its goals than a periodic timer: the push server receives “connection availability information” (e.g., from a device with “higher bandwidth capacity,” such as a broadband access point) and “uses the connection availability information as a trigger event for sending lazy or low priority messages ... routed through the higher bandwidth capacity device to the wireless terminal.” Ex. 1007, 14, 33-34 (claims 24-28). In other words, Houghton transfers non-time-critical data according to the availability of high-efficiency transfer options—not according to the expiration of a timer. A POSITA would understand that delivering such data when an efficient path is actually available accomplishes Houghton’s cost-efficiency goal directly, whereas a periodic timer would fire without regard to whether an efficient path exists at that moment.

147. Nothing in Houghton, Munson, or Adamczyk counsels in favor of departing from these approaches. As discussed above, neither Houghton nor Munson discloses triggering message delivery based on the expiration of a periodic timer at all, much less any benefit of doing so. Adamczyk mentions sending notification messages “on a periodic schedule” in passing, as one of several options—“the notification server is configured to send the notification messages to the recipients on demand, at a specific future time, and/or on a periodic schedule” (Ex. 1021, [0010])—with no accompanying description of any benefit. In my opinion, this passing mention would not have motivated a POSITA to deviate from

Houghton’s express design goals of (1) rapid transmission for time-critical messaging and (2) cost-efficient, availability-triggered transmission for non-time-critical messaging.

148. Finally, Dr. Traynor’s proposed implementation—configuring the push server “to hold received messages in the buffer until expiry of a recurring timer before pushing messages received during the time interval at once” (Ex. 1003, ¶926)—would introduce its own problems. A POSITA would understand that releasing all messages accumulated over an interval simultaneously can produce large, bursty transfers that stress the network connection and the receiving device—particularly for the non-time-critical content contemplated by Houghton (application code and data updates), which involves larger data transfers. Neither Dr. Traynor nor the references explain why a user would prefer to receive accumulated messages in periodic bursts in the context of Houghton’s system, or how doing so would be consistent with Houghton’s design goals. And this would be additionally problematic in the context of Houghton’s “push-type command of applications” (e.g., Ex. 1007 at 11), wherein sending such messages periodically would predictably result in multiple applications being commanded to do multiple different things simultaneously, leading to a negative user experience.

149. For all of these reasons, it is my opinion that the Houghton-Munson-Adamczyk theory fails as to claim 3, and, because claim 4 depends from claim 3, as to claim 4 as well.

E. Claim 17: Houghton-Munson-Fok.

150. Dr. Traynor fails to establish the obviousness of claim 17 in his Ground 2G for the same reason he fails to establish the obviousness of claim 17 in his Ground 1G, my response to which is incorporated by reference.

1. Dr. Traynor fails to show it would be obvious to combine Houghton-Munson with Fok in his proposed manner.

151. Neither Fok nor Houghton-Munson discloses a two-handshake process to deliver a single message; in Fok, applications communicate directly with one another. This, however, would not satisfy the claim requirements, which is presumably why Dr. Traynor does not propose modifying the Houghton-Munson device with what is actually disclosed in Fok.

152. In my opinion, Dr. Traynor's deviation from Fok's single-handshake mechanism to implement a two-handshake procedure with Houghton's push client serving as a middleman would not have been obvious. I disagree with Dr. Traynor's stated rationales in this regard. For example, Dr. Traynor alleges that his proposed deviation from Fok's teachings would (1) allow applications on a single device "to interact with one another without needing to develop, for instance, custom APIs ('Application Programming

Interfaces’) for alternative forms of direct communication,” and (2) “ensure the applications were trusted before sharing information; the push client would provide the interface and relevant authorizations.” Ex. 1003, ¶980.

153. Regarding the “custom APIs” aspect of Dr. Traynor’s opinion, Dr. Traynor presents no evidence that in a device implementing Houghton-Munson, *direct* communication between applications (rather than using the push client as a middleman as Dr. Traynor proposes) would require “custom APIs” above and beyond what would be required for Dr. Traynor’s proposal (which deviates from the teachings of Fok). But there is no reason that the *same* API already used to transmit data to the push client could not be used to transmit data to other applications directly, and Dr. Traynor identifies none. For example, as Dr. Traynor acknowledges, “Houghton-Munson does not expressly describe inter-application transmission on a given device” such that it would not require any particular protocol for transferring data between the push client and individual applications, such that the API used for such a transmission need not be exclusive to push transmissions or any particular push protocol. Ex. 1003, ¶978. Thus, to the extent the applications used APIs for communication at all, there is no reason that the *same* API used for communication between the push client and other applications could not be used for communication with individual applications residing on a device, such

that no new “custom APIs” would need to be developed “for alternative forms of direct communication” as Dr. Traynor contends. And certainly, there is no evidence that a POSITA would have recognized development of custom APIs as a problem that would require deviation from Fok’s teaching of applications directly communicating with multiple different applications within a device. *See* Ex. 1038, [0047] (noting that a given application “can maintain multiple secure communication sessions with multiple other applications... concurrently”).

154. Regarding Dr. Traynor’s allegation that his proposed deviation from Fok’s teachings would “ensure the applications were trusted before sharing information” because “the push client would provide the interface and relevant authorizations” (Ex. 1003, ¶980), Dr. Traynor does not explain how this would solve any problem in the direct application-to-application handshake mechanism that Fok already discloses. For example, as noted above, applications would ***already*** need a communication interface for communicating with the push client, and Fok already teaches that a single application can communicate with multiple other applications, such that there is no reason why using Fok’s handshake teachings ***without*** inserting the push client as a middleman would have required a separate interface. And individual applications would ***already*** need to be configured to perform authorization

functionality under Dr. Traynor’s proposal, in order to engage with Fok’s handshake mechanism (even if transmission was limited only to the push client as Dr. Traynor proposes). Dr. Traynor is thus incorrect that the push client would perform all relevant authorizations under his proposed combination—the individual applications would *also* need to be configured to provide authorization functionality, regardless of whether the push client was used as a middleman component as Dr. Traynor proposes. *See, e.g.*, Ex. 1038, [0006] (noting that a “recipient application... reference[s] the unique identifier with a list of certified primary applications to verify the primary application”); *id.*, [0052] (discussing applications checking a unique ID to determine whether a primary application is “an un-authorized application”).

155. Nor is the use of the push client as a middleman necessary to “ensure the applications were trusted before sharing information” as Dr. Traynor implies. As Fok discloses, typically, “an operating system (106) can require certain standards be met for applications designed to execute on the operating system (106),” including “incorporat[ing] a list of unique identifiers of trusted primary applications” such that a “recipient application... could then verify that a received handshake request is sent by an application authorized to make the request” and the recipient application “could then trust and engage in communication with such application.” Ex. 1038, [0048]. In other words, Fok

already teaches techniques to ensure that applications are trusted before sharing information, and the introduction of an unnecessary push client middleman in the transmission process thus does not solve a problem in Fok.

156. Furthermore, I am aware of no evidence that implementing a handshake technique as described in Fok to require *two* handshakes for a single transmission between two applications (as opposed to one handshake, as proposed by Fok itself) was conventional. Nor do I believe this modification would have been obvious. For example, Dr. Traynor presents no evidence that a POSITA would have found Fok's technique of direct communication between applications residing on a device to be deficient. Nor do I believe that a POSITA would have viewed Dr. Traynor's proposed deviation from Fok's teachings as an improvement over Fok's teachings, even if a POSITA was for some reason looking to improve upon Fok's teachings. A transmitting application must know which application it intends to receive the message, because under Dr. Traynor's proposal each message must carry an identifier for the recipient application. *See* Ex. 1003, ¶915 (opining that, in Houghton-Munson, "each such message intended for an application executing on one of the mobile terminals would include data and corresponding identification of the intended software application"). It follows that the transmitting application would be able to address the recipient application directly using Fok's

handshake technique as disclosed, without using the push client as an unnecessary middleman with an additional handshake procedure.

157. I also disagree with other of Dr. Traynor's statements that could be viewed as alleged motivations to deviate from Fok's disclosures. For example, Dr. Traynor alleges his deviation from Fok's teachings would be advantageous because "each application would not need to be configured with a brand-new 'handshake' mechanism, but would instead utilize its existing protocols for Houghton-Munson's transmissions." Ex. 1003, ¶981. What Dr. Traynor's opinion ignores is that an application's "existing protocols for Houghton-Munson's transmissions" do not include handshake procedures at all. *See generally* Exs. 1007, 1017 (no discussion of handshakes for communicating with the push client or other applications on a device). Thus, a POSITA attempting to implement **any** usage of Fok's handshake techniques for intra-device communication would need to configure each participating application with "a brand-new handshake" mechanism (i.e., Fok's handshake mechanism that Dr. Traynor himself proposes would be used). And Dr. Traynor presents no reason why having applications communicate with each other directly (as opposed to communicating with the push client) would require a **distinct** handshake mechanism. It would not, as Fok makes clear that its handshake mechanism allows any participating application "can maintain

multiple secure communication sessions with multiple other applications... concurrently” (Ex. 1038, [0047]), and a POSITA would have understood that the *same* handshake mechanism would be used for each of these concurrent sessions (as evidenced by Fok containing no disclosure that multiple distinct handshake protocols would be required for multiple concurrent sessions).

158. Furthermore, Dr. Traynor’s proposal would have the obvious drawback of doubling the number of handshakes and communications required for communication between two applications. For instance, under Fok’s disclosure, under Dr. Traynor’s first example where a “messaging application could direct a video-playback application to download a specific codec,” it would require the messaging application to initiate a handshake with the push client and further require the push client to subsequently initiate a handshake with the video-playback application in order to secure the message delivery. There is absolutely no evidence that a POSITA would have been motivated to implement this solution, and in my opinion a POSITA would have avoided doing so because it would require the number of communication sessions and handshakes to be doubled, increasing transmission time, processing power, and battery drain. *See* Ex. 1007, 20 (noting that “protocol overhead such as... handshake” should ideally “occur[] before the actual need for message transmission” because handshaking at the time of transmission would “delay[]

the transmission,” such that a POSITA would have avoided unnecessary handshakes because they lead to delay).

159. Another disadvantage to Dr. Traynor’s proposal (relative to just implementing Fok’s teachings within Houghton-Munson without modifying them to involve the push client as an intermediary) is that it assumes that communicating applications would already have an API for communication with a push client. *See* Ex. 1003, ¶980 (alleging that without his proposed combination, alternative custom APIs would be necessary for communication). His proposal would thus only work for applications that already communicate with the push client. But to the extent a POSITA would have recognized secure interprocess communication within a device implementing Houghton-Munson as an advantage in the first place, there is no reason they would not design the solution to allow for secure interprocess communication between all applications (not merely applications that already interface with the push client). For example, a POSITA would recognize, to use Dr. Traynor’s example, that it would be common for an application such as a “[photo]-editing application” (Ex. 1003, ¶980) to not interact with the push client, but still benefit from communication with other processes within the end-user device. Dr. Traynor’s proposal to use the push client as a middleman would not allow such an application to communicate with other applications such as a

messaging application, but implementing Fok's techniques to allow for direct communication between photo-editing application and messaging application *would* allow for the communication to occur, and would further accomplish that result with less signaling overhead.

2. Dr. Traynor does not establish that his proposal, even if obvious, satisfies the claim requirements.

160. Claim 17 requires that the device link agent "caus[es] messages to be securely delivered to a software process by initiating delivery of each such message on the secure interprocess communication service." Thus, messages must be delivered on an "interprocess communication service," and that service itself must be "secure."

161. Dr. Traynor does not identify *any* alleged "service" in either Houghton-Munson or Fok that he contends would constitute a claimed "interprocess communication service." Ex. 1003, ¶¶977-983 (not using the term "service" once except in restating the claim language, and not identifying anything in Fok or Houghton-Munson that allegedly constitutes the claimed "service"). As best I can understand, Dr. Traynor's contention is that secure delivery to a software process inherently involves a "secure interprocess communication service." Ex. 1003, ¶977 (discussing "a device link agent... on a given device causing messages to be securely delivered to a software

process” as allegedly satisfying the “secure interprocess communication service” requirement). But as discussed above in Section VI.C regarding claim construction, interprocess communication need not involve an interprocess communication *service*, just as two individuals communicating with one another need not involve a communication service such as telephone or mail service.

162. Furthermore, Dr. Traynor would *additionally* need to show that any “interprocess communication service” identified is itself “secure.” But Dr. Traynor does not explain how any *communication service* used for communication is itself secure. Indeed, Fok itself seems to assume that to the extent any service is used for communication, that service is not itself secure, such that applications must identify their *own* form of security through Fok’s handshake process.

163. Thus, Fok discloses secure communication via a handshake process. But this handshake process is not a “communication service” provided to the applications—instead, it is a security mechanism implemented by each individual application and coordinated between applications. For example, consider an unsecured public Wi-Fi connection over a Starbucks Wi-Fi hotspot. The Wi-Fi hotspot provides a “communication service” between a user’s laptop and the Internet, but that “communication service” is not secure

because it involves a public hotspot that does not provide a secure connection. Nevertheless, if messages transmitted over that unsecured communication service are secured via encryption mechanisms, secure delivery of messages may still be possible over the unsecured communication service. The fact that it is possible to securely transmit messages over an unsecured communication service does not transform an unsecured Wi-Fi connection into a “secure communication service”; rather, it emphasizes that any security of transmission comes not from the communication service itself but from security mechanisms (such as encryption protocols) handled between the sending and receiving entities.

164. In the same manner, a POSITA would understand that even if Dr. Traynor *had* identified some “communication service” in Fok or Houghton-Munson that provides communications between processes, any security of his proposal comes from handshaking mechanisms implemented by the applications themselves rather than security that is derived from whatever “communication service” is used to deliver messages. In my opinion, this is insufficient to satisfy claim 17, which requires that the communication service itself be secure (rather than the processes communicating over the communication service provide additional security mechanisms that are not themselves part of the provided interprocess communication service).

165. In short, Dr. Traynor neither identifies an “interprocess communication service” nor explains how any such service would itself be “secure” as the claims require.

XII. CONCLUSION

166. For the reasons set forth above, it is my opinion that: (1) under Dr. Traynor’s own interpretation of “message delivery trigger,” the receipt of every received network element message in TS-23.140 and in the Houghton/Munson combination is a message delivery trigger, such that neither satisfies the requirement that “for at least some of the received network element messages, the receipt of such a message by the message buffer system is not a message delivery trigger”; (2) under the plain and ordinary meaning of “message delivery trigger,” TS-23.140 discloses a single message delivery trigger—reception of a valid retrieval request—and not the claimed “plurality”; (3) none of Dr. Traynor’s theories establishes that “one of the message delivery triggers is the expiration of a periodic timer,” as claim 3 requires, under the plain and ordinary meaning of “periodic timer”; (4) a POSITA would not have been motivated to combine Houghton with Munson, or to modify TS-23.140 or Houghton/Munson in view of Adamczyk, in the manners Dr. Traynor proposes, nor would a POSITA have had a reasonable expectation of success in doing so; and (5) a POSITA would not have been motivated to modify the prior art in the manner Dr. Traynor proposes to

arrive at claim 17, and his proposals would not satisfy the claim language in any event. Because every challenged claim of the '320 patent depends, directly or indirectly, from claim 1, the deficiencies I identify with respect to limitation [1d3] apply to all of the challenged claims. I reserve the right to supplement my opinions to address any new information or argument, including any new positions taken by Petitioners or Dr. Traynor.

167. 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.

August 4, 2026



Michael C. Brogioli, Ph.D.