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.

IPR2026-00106

U.S. Patent No. 10,321,320

PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,321,320

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

Google Exhibit 1080
Google v. Headwater
IPR2026-00271

TABLE OF CONTENTS

I. IPR REQUIREMENTS 1
A. Grounds for Standing 1
B. Identification of Challenge 1
1. Prior Art 1
2. Grounds for Challenge 3
II. ’320 PATENT 4
III. LEVEL OF ORDINARY SKILL IN THE ART 5
IV. CLAIM CONSTRUCTION 5
V. THE CHALLENGED CLAIMS ARE UNPATENTABLE 6
A. Ground 1A: TS-23.140 (Claims 1-3, 5-7, 10-11, 15, 18) 6
1. TS-23.140 6
2. Claim 1 7
3. Claim 2 26
4. Claim 3 26
5. Claims 5-6 27
6. Claim 7 28
7. Claim 10 28
8. Claim 11 28
9. Claim 15 30
10. Claim 18 32
B. Ground 1B: TS-23.140-Adamczyk (Claim 3) 33
C. Ground 1C: TS-23.140-Herzog (With/Without Adamczyk) (Claim 4) 35
D. Ground 1D: TS-23.140-Shen (Claims 6, 12-13, 16) 37
1. Shen 37
2. Claim 6 38
3. Claim 12 39

4. Claim 13 41
5. Claim 16 41
E. Ground 1E: TS-23.140-Pazhyannur (Claims 8-9) 42
F. Ground 1F: TS-23.140-Ellison (Claim 14) 44
G. Ground 1G: TS-23.140-Fok (Claim 17) 47
H. Ground 2A: Houghton-Munson (Claims 1-7, 10-11, 15, 18) 49
1. Houghton 49
2. Munson 50
3. Houghton-Munson Combination 51
4. Claim 1 53
5. Claim 2 75
6. Claim 3 76
7. Claim 4 77
8. Claims 5-6 78
9. Claim 7 79
10. Claim 10 80
11. Claim 11 80
12. Claim 15 81
13. Claim 18 82
I. Ground 2B: Houghton-Munson-TS-23.140 (Claims 1-7, 10-11, 15, 18) 83
J. Ground 2C: Houghton-Munson-Adamczyk (Claims 3-4) 85
K. Ground 2D: Houghton-Munson-Shen (Claims 6, 12-13, 16) 86
1. Claim 6 86
2. Claim 12 87
3. Claim 13 89
4. Claim 16 89

L. Ground 2E: Houghton-Munson-Pazhyannur (Claims 8-9) 90
M. Ground 2F: Houghton-Munson-Ellison (Claim 14) 91
N. Ground 2G: Houghton-Munson-Fok (Claim 17) 93
VI. CONCLUSION 94
VII. MANDATORY NOTICES 94
A. Real Party in Interest 94
B. Related Matters 94
C. Notice of Counsel and Service Information 95
D. Power of Attorney 96

EXHIBIT LIST

No.	Exhibit
1001	U.S. Patent No. 10,321,320 ("'320 patent" or "'320Pat")
1002	File History of the '320 patent ("'320FH")
1003	Declaration and Curriculum Vitae of Dr. Patrick Traynor ("Traynor")
1004	3GPP TS 23.140 v6.9.0 (2005-03); 3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional Description; Stage 2 ("TS-23.140")
1005	U.S. Patent Pub. No. 2006/0190720 to Ozaki et al. ("Ozaki")
1006	WO 2008/048075 to Lee et al. ("Lee")
1007	WO 2006/077283 to Houghton et al. ("Houghton")
1008	U.S. Patent Pub. No. U.S. 2009/0158397 to Herzog et al. ("Herzog")
1009	U.S. Patent No. U.S. 7,925,717 to Chou et al. ("Chou")
1010	Open Mobile Alliance; Multimedia Messaging Service Architecture Overview (MMSARCH) specification, July 15, 2004, *available at* https://www.openmobilealliance.org/release/MMS/V1_1-20040715-A/OMA-WAP-MMS-ARCH-V1_1-20040715-A.pdf
1011	Open Mobile Alliance; OMA-ERELD-MMS-v1_2-20030923-C, Enabler Release Definition for MMS Version 1.2, Sept. 23, 2003, *available at* https://www.openmobilealliance.org/release/MMS/V1_2-20030923-C/OMA-ERELD-MMS-V1_2-20030923-C.pdf
1012	U.S. Patent No. U.S. 7,509,487 to Lu et al. ("Lu")

No.	Exhibit
1013	Technical Specification Group Services and System Aspects Meeting #19, TSGS#19(03)0167, European Telecommunications Standards Institute February 2003, Mar. 12, 2003, *available at* https://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_19/Docs/PDF/SP-030167.pdf
1014	U.S. Patent Pub. No. U.S. 2005/0207379 to Shen et al. ("Shen")
1015	RESERVED
1016	Declaration of Friedhelm Rodermund
1017	U.S. Patent Pub. No. U.S. 2009/0240807 to Munson et al. ("Munson")
1018	EP Patent Application EP1853044 to Shenfield ("Shenfield")
1019	U.S. Patent No. U.S. 7,082,615 to Ellison et al. ("Ellison")
1020	RESERVED
1021	U.S. Patent Pub. No. 2008/0162637 to Adamcyzk et al. ("Adamcyzk")
1022	Dismissal with Prejudice, *Headwater Research LLC v. Samsung Electronics Co.*, Case No. 2:23-cv-00103 (E.D. Tex., May 1, 2025), ECF No. 438
1023	Claim Construction Order, *Headwater Research LLC v. Samsung Electronics Co.*, Case No. 2:23-cv-00103 (E.D. Tex., Aug. 22, 2024), ECF No. 118
1024	Gang Lu, et al., *Heading for Multimedia Message Service in 3G*, 6th IEE International Conference on 3G and Beyond, Washington, D.C., USA, Nov. 7-9, 2005
1025	RFC 4355, IANA Registration for Enumservices Email, Fax, MMS, EMS, and SMS (Jan. 2006)

No.	Exhibit
1026	Friedhelm Rodermund, *A Picture Speaks a Thousand Words – From SMS to MMS*, in *Business Briefing: Wireless Technology* (2003)
1027	IETF RFC 793, Transmission Control Protocol (Sept. 1981), *available at* https://www.ietf.org/rfc/rfc793.txt
1028	The TLS Protocol Version v 1.0 (Jan. 1999), *available at* https://datatracker.ietf.org/doc/html/rfc2246
1029	Complaint for Patent Infringement, *Headwater Research LLC v. Samsung Electronics Co.*, Case No. 2:23-cv-00103 (E.D. Tex. Mar. 10, 2023), ECF No. 1
1030	Roger M. Needham & Michael D. Schroeder, *Using Encryption for Authentication in Large Networks of Computers*, ACM, Vol. 21, No. 12 (Dec. 1978) ("Needham")
1031	Michael D. Schroeder & Jerome H. Saltzer, *A Hardware Architecture for Implementing Protection Rings ACM*, Vol. 15, No. 3 (Mar. 1972) ("Schroeder")
1032	Jerome H. Saltzer & Michael D. Shroeder, *The Protection of Information in Computer Systems IEEE Proceedings*, Vol. 63, No. 9 (Sept. 1975) ("Saltzer")
1033	Bo Li et al., *Symbian OS platform security model*, in *Login Magazine* (Aug. 2010) *available at* https://www.usenix.org/system/files/login/articles/73507-li.pdf ("Li")
1034	Philip Zimmermann, "Pretty Good Privacy: RSA Public Key Cryptography for the Masses," PGP User's Guide, Version 1.0, (June 1991), *available at* https://www.techinsider.org/freesoftware/research/acrobat/910605.pdf ("Zimmerman")
1035	B. Ramsdell, *S/MIME Version 3 Message Specification*,

No.	Exhibit
	IETF RFC 2633, (June 1999), *available at* https://datatracker.ietf.org/doc/html/rfc2633 ("Ramsdell")
1036	Miraj E. Mostafa, *Transporting data between wireless applications using a messaging system—MMS (Wireless Comms. and Mobile Computing* (July 7, 2006) ("Mostafa")
1037	U.S. Patent Pub. No. 2008/0243999 to Pazhyannur et al. ("Pazhyannur")
1038	U.S. Patent Pub. No. 2008/0215883 to Fok et al. ("Fok")
1039	U.S. Patent Pub. No. 2003/0126282 to Sarkar et al. ("Sarkar")
1040	U.S. Patent Pub. No. 2007/0037610 to Logan ("Logan")
1041	U.S. Patent Pub. No. 2005/0091380 to Gonen et al. ("Gonen")
1042	U.S. Patent Pub. No. 2004/0258063 to Raith et al. ("Raith")
1043	U.S. Patent Pub. No. 2004/0085894 to Wang et al. ("Wang")
1044	U.S. Patent No. 6,094,424 to Kalmanek et al. ("Kalmanek")
1045	Simon Higginson, *Platform Security Concepts*, *in* SYMBIAN OS PLATFORM SECURITY: SOFTWARE DEVELOPMENT USING THE SYMBIAN OS SECURITY ARCHITECTURE, 17, 17-41 (Craig Heath ed., 2006) ("Higginson")
1046	U.S. Patent Pub. No. 2003/0193967 to Fenton et al. ("Fenton")
1047	U.S. Patent Pub. No. 2004/0111476 to Trossen et al. ("Trossen")
1048	U.S. Patent Pub. No. 2005/0282531 to Andreasson et al. ("Andreasson")

No.	Exhibit
1049	U.S. Patent No. 9,615,192 to Raleigh ("'192 patent" or "'192Pat")

LISTING OF CHALLENGED CLAIMS

Claim	Limitation
Claim 1	
1[pre]	A networked system comprising:
1[a]	i) a network server system including
1[b1]	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,
1[b2]	each of the wireless end-user devices comprising multiple software components authorized to receive messages via the device link agent on that device;
1[c1]	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,
1[c2]	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
1[d1]	a message buffer system including a memory and logic,
1[d2]	the memory to buffer content from the received network element messages for which delivery is requested to any of the wireless end-user devices,
1[d3]	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

Claim	Limitation
1[d4]	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 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
1[e1]	(ii) the device link agents on the respective wireless end-user devices, each of the device link agents configured to
1[e2]	maintain the respective secure Internet data message link over a wireless network to the link interface,
1[e3]	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
1[e4]	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.
Claim 2	
2	The networked system of claim 1, the message buffer system logic to determine, for each of the wireless end-user devices, when one of a plurality of message delivery triggers has occurred,

Claim	Limitation
	at least one of the triggers for each given device specific to one or more states of that given device.
Claim 3	
3	The networked system of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.
Claim 4	
4	The networked system of claim 3, wherein 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.
Claim 5	
5	The networked system of claim 1, wherein one of the message delivery triggers is the receipt of a transmission on the respective secure Internet data message link from the device link agent of the given one of the wireless end-user devices, or a response generated to a transmission received from that device link agent.
Claim 6	
6	The networked system of claim 1, wherein one of the message delivery triggers is a heartbeat message generated by the given device link agent, or a request received from the given device link agent.
Claim 7	
7	The networked system of claim 1, wherein one of the message delivery triggers is the receipt of a particular message from one of the network elements.
Claim 8	
8	The networked system of claim 1, further comprising a secure server to store a secure authorization list, the secure authorization

Claim	Limitation
	list indicating the authorized software components and the authorized network elements that are allowed to communicate using the network server system.
Claim 9	
9	The networked system of claim 8, further comprising the respective device link agent on each wireless end-user device receiving access authorization information from the secure server, the access authorization indicating, respectively for each wireless end-user device, the software components authorized to receive messages via the device link agent on that device.
Claim 10	
10	The networked system of claim 1, wherein a second subset of the secure Internet data messages contain a user message from a network element, the user message intended for display on a user interface of a given one of the wireless end-user devices.
Claim 11	
11	The networked system of claim 1, wherein one of the software components on a given one of the wireless end-user devices is a policy control agent, and the data to be consumed by the policy control agent comprises service settings and/or configuration information for the given device.
Claim 12	
12	The networked system of claim 1, the link interface to encrypt messages identified for delivery to each given one of the wireless end-user devices to create secure Internet data messages, the device link agent on each given device further configured to decrypt the received secure Internet data messages for that device prior to delivering those messages to a respective software process.

Claim	Limitation
Claim 13	
13	The networked system of claim 12, wherein the encrypted messages are transported to the device link agent using one or more of encryption on a transport services stack, IP (Internet Protocol) layer encryption, and tunneling.
Claim 14	
14	The networked system of claim 1, wherein the device link agent on a given device executes in a secure execution environment, and at least one of the software components on that device executes outside of the secure execution environment.
Claim 15	
15[pre]	The networked system of claim 1,
15[a]	wherein the device link agent on a given one of the devices is further configured to receive, from one or more of the software components on that device, upload messages,
15[b]	each of the upload messages identifying a corresponding one of the network elements to which the device respective software component has requested delivery,
15[c]	the device link agent on that device transmitting the upload messages to the network message server over the respective secure Internet data message link, for delivery by the network message server to the respective identified network elements.
Claim 16	
16	The networked system of claim 15, the device link agent on the given device further configured to buffer one or more of the upload messages for transmission to the network message server at a time selected by a heartbeat mechanism.

Claim	Limitation
Claim 17	
17	The networked system of claim 1, at least a given one of the devices further comprising a secure interprocess communication service separately secured from the secure Internet data message link, the device link agent for the given device causing messages to be securely delivered to a software process by initiating delivery of each such message on the secure interprocess communication service.
Claim 18	
18	The networked system of claim 1, wherein at least one of the secure Internet data messages comprises multiple identifier/data pairs.

Amazon.com Services LLC and Amazon Web Services, Inc. ("Amazon" or "Petitioners") petition for *inter partes* review ("IPR") of claims 1-18 ("Challenged Claims") of U.S. Patent 10,321,320 ("'320Pat").

I. IPR REQUIREMENTS

A. Grounds for Standing

Petitioners certify the '320Pat is available for IPR, and Petitioners are not barred or estopped from requesting IPR on the grounds herein.

B. Identification of Challenge

1. Prior Art

Petitioners' grounds rely upon the prior art below based on an assumed priority date of January 28, 2009.[1] The references are analogous to the '320Pat and each other for being in the same field of endeavor (e.g., computer systems and/or networks) or reasonably pertinent to the problems faced by the inventor (e.g., computer messaging across a network and/or network/device security). Traynor, ¶99.

[1] Petitioners do not concede entitlement to this priority date.

Prior Art	Date	Basis[2]
TS-23.140 (Ex-1004)	Publicly available/accessible on/around April 2005[3]	102(b)
Houghton (Ex-1007)	Published 7/27/2006	102(b)
Munson (Ex-1017)	Filed 3/21/2008	102(e)
Shen (Ex-1014)	Published 9/22/2005	102(b)
Ellison (Ex-1019)	Published 7/25/2006	102(b)
Pazhyannur (Ex-1037)	Published 10/2/2008	102(a)/(e)
Herzog (Ex-1008)	Filed 12/17/2007	102(e)
Fok (Ex-1038)	Published 9/4/2008	102(a)/(e)
Adamczyk (Ex-1021)	Published 7/3/2008	102(a)/(e)

[2] Citations are to the pre-AIA statute; if the post-AIA statute applies, the analysis would be identical.

[3] Ex-1016 (public availability/accessibility by April 4, 2005, and similar for Open Mobile Alliance ("OMA") references).

2. Grounds for Challenge

Ground (all §103)	Claims	Prior Art
1A	1-3, 5-7, 10-11, 15, 18	TS-23.140
1B	3	TS-23.140-Adamczyk
1C	4	TS-23.140-Herzog (with/without Adamczyk)
1D	6, 12-13, 16	TS-23.140-Shen
1E	8-9	TS-23.140-Pazhyannur
1F	14	TS-23.140-Ellison
1G	17	TS-23.140-Fok
2A	1-7, 10-11, 15, 18	Houghton-Munson
2B	1-7, 10-11, 15, 18	Houghton-Munson-TS-23.140
2C	3-4	Houghton-Munson-Adamczyk (with/without TS-23.140)
2D	6, 12-13, 16	Houghton-Munson-Shen (with/without TS-23.140)
2E	8-9	Houghton-Munson-Pazhyannur (with/without TS-23.140)
2F	14	Houghton-Munson-Ellison (with/without TS-23.140)
2G	17	Houghton-Munson-Fok (with/without TS-23.140)

II. ’320 PATENT

The ’320Pat’s “networked system” uses a “buffer” to store messages (like text/multimedia messages) from network elements and delivers the messages to end-user devices when some triggering event occurs. ’320Pat, Abstract; Traynor, ¶¶49-66, 91-98.

To transmit messages to end-user devices, a secure link is established between the “networked system” (red) and a “device link agent” (blue) on the end-user’s device. ’320Pat, Abstract, 89:11-90:4; Traynor, ¶92.

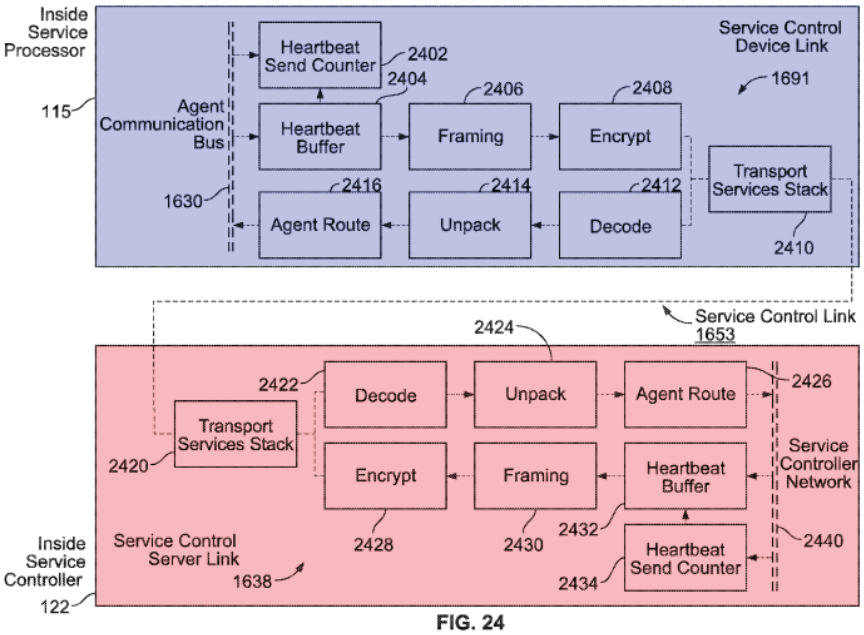


FIG. 24

’320Pat, Fig. 24.[4]

[4] Color annotations added unless otherwise noted.

III. LEVEL OF ORDINARY SKILL IN THE ART

Persons of ordinary skill in the art ("POSITAs") relating to the '320Pat's subject matter 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) 3-5 years of experience in services and application implementation in communication networks. Traynor, ¶¶47-48. Additional graduate education could substitute for professional experience, and vice versa. *Id.*

IV. CLAIM CONSTRUCTION

No express constructions are necessary in this IPR; Petitioners have applied the claim terms' ordinary meanings as understood by POSITAs.[5] 37 C.F.R. §42.100(b). If the '320Pat specification/'320FH inform certain terms' ordinary meanings, Petitioners address this in the grounds below.

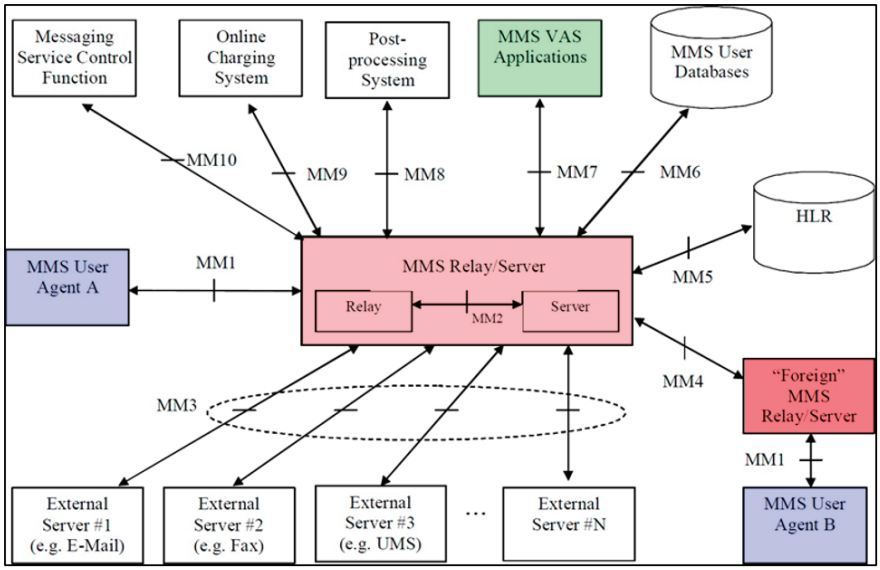
[5] Petitioners reserve the right to respond to PO/Board constructions. Petitioners reserve §112/claim scope arguments.

V. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Ground 1A: TS-23.140 (Claims 1-3, 5-7, 10-11, 15, 18)

1. TS-23.140

TS-23.140 is a technical specification issued by the 3GPP standards body to standardize multimedia transmissions through a "Multimedia Messaging Service" ("MMS"). TS-23.140, Figs. 3, 10, 23[6]; Traynor, ¶¶100-07.



TS-23.140, Fig. 23.

An "MMS Relay/Server" ("Relay/Server") (light red) coordinates storage, notification, reporting, and handling of multimedia messages ("MM[s]"). *Id.*, 21, 23-24. To transmit messages, Relay/Server may coordinate between MMS User Agents

[6] Citations refer to publication page number.

("User Agents") on end-user devices (blue), out-of-network Relay/Servers (dark red), and Value Added Services ("VAS") Applications (green). *Id.*

MMS messages may include "data specific to applications between two MMS User Agents or an MMS User Agent and an MMS VAS Application (or vice versa)." *Id.*, 54-55.

2. Claim 1

a. 1[pre]-1[a]

TS-23.140 discloses/suggests 1[pre] (if limiting)/1[a]. Traynor, ¶¶539-47.

TS-23.140's network of servers, User Agents, databases, and applications (*networked system*[7]) includes (i) Relay/Server (*network server system*), which facilitates message delivery over the *networked system*; and (ii) User Agents (*device link agents*) executing on user equipment ("UE")/mobile devices. TS-23.140, 14, 17, 19-20, 54-55; §§V.A.2.b-V.A.2.l (1[b1]-[e4]); Traynor, ¶¶540-47.

[7] Italics indicate claim language.

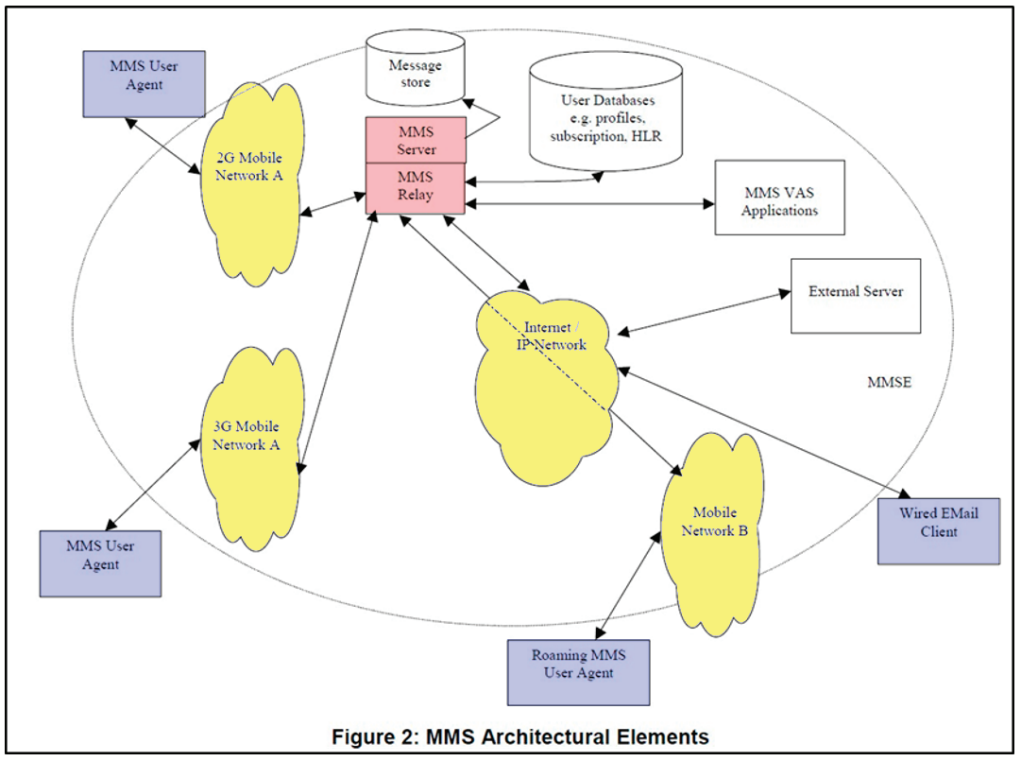


Figure 2: MMS Architectural Elements

TS-23.140, Fig. 2.

b. 1[b1]

TS-23.140 discloses/suggests 1[b1]. Traynor, ¶¶548-61.

TS-23.140's Relay/Server establishes/maintains a link over an Internet network (yellow) with User Agents executing on each UE/mobile device. TS-23.140, 17, 19-20; Traynor, ¶¶549-50; Ex-1010, Fig. 3.

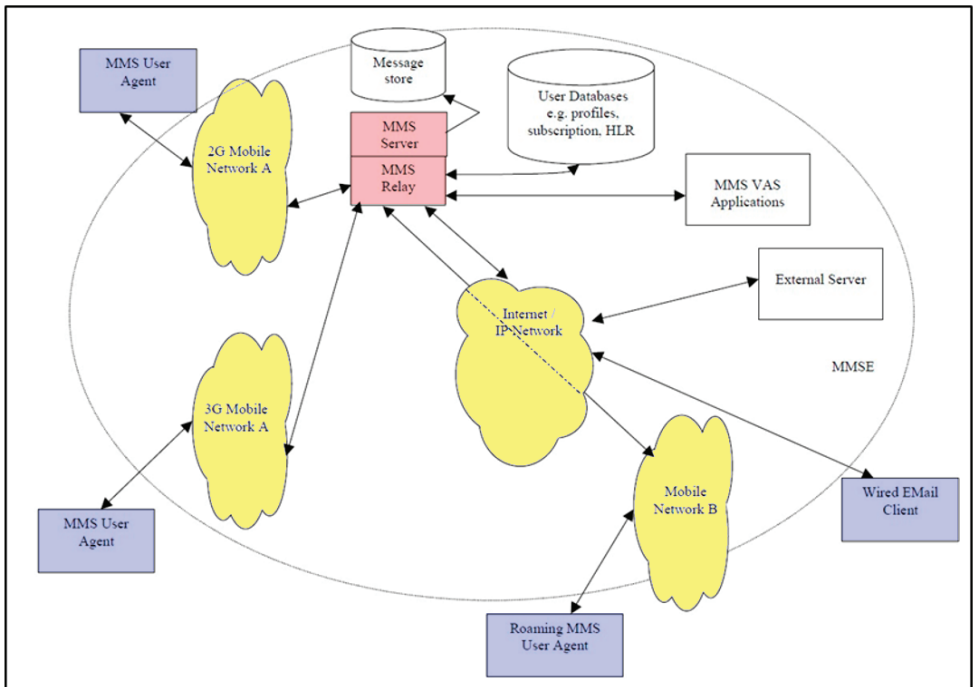
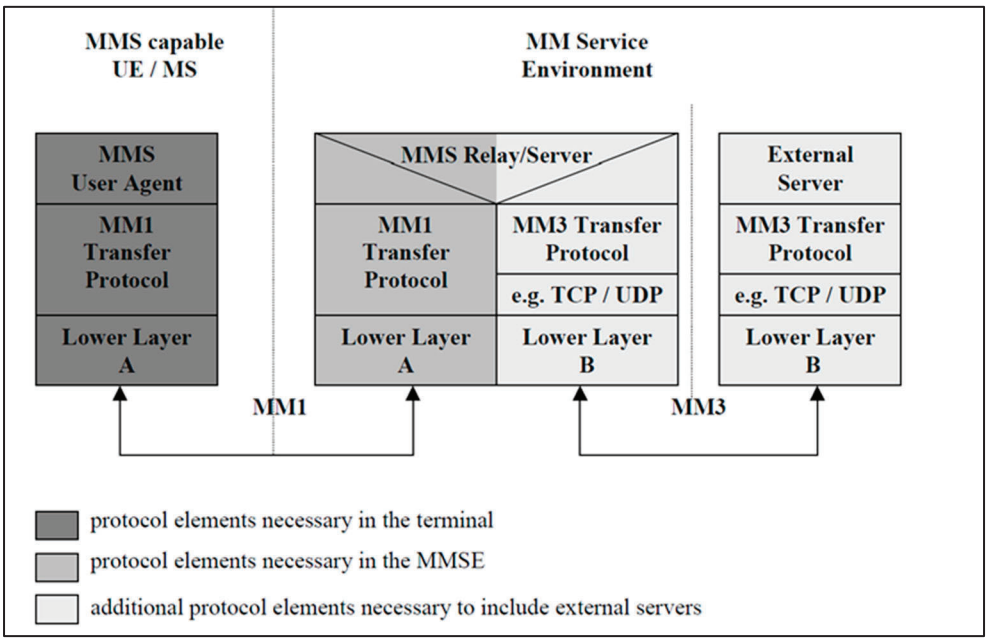


Figure 2: MMS Architectural Elements

TS-23.140, Fig. 2.

In TS-23.140, (1) Relay/Server and User Agent utilize a link interface including MM1 Transfer Protocol to transmit messages over the Internet between Relay/Server and User Agents on UEs/mobile devices; and (2) MM1 Transfer Protocol includes transmission-control protocols ("TCP") and transport-layer security ("TLS"), which secures/encrypts network communications within transport stacks. TS-23.140, 24-25, 41; Traynor, ¶¶551-59 (citing Shen, [0017]).



TS-23.140, Fig. 4.

In TS-23.140, MM1 Transfer Protocol may implement the "Wireless Application Protocol" ("WAP") defined by the Open Mobile Alliance ("OMA") and incorporates OMA's specifications. TS-23.140, 13, 162; Ex-1011.

POSITAs would have known "a device implementing OMA MMS ***must have…WAP WSP stack or HTTP/TCP/IP stack***."[8] Ex-1011, 11. Further, OMA has TLS protocols for "secure data transmission between the MMS Client and the MMS Proxy-Relay in…HTTP based protocol stacks for MMS implementation." Ex-1010, 22; Traynor, ¶¶555-58.

[8] Emphases added unless otherwise noted.

POSITAs would have understood/found obvious TS-23.140 discloses a *link interface* comprising MM1 Transfer Protocol with TCP/IP and TLS protocols. Traynor, ¶559.

POSITAs would have also understood/found obvious that, in MMS environments, multiple User Agents communicate with Relay/Server and maintain respective TLS-based links between each User Agent and Relay/Server. Traynor, ¶560 (citing Mostafa, 2-3, Figs. 1, 3; Munson, Fig. 1, [0007]-[0008]; Houghton, 23).

Accordingly, TS-23.140 discloses/suggests Relay/Server (*network server system*) comprises MM1 Transfer Protocol with TCP/IP and TLS (*link interface*) to communicate with UEs/mobile devices. Traynor, ¶¶555-61. Further, TS-23.140 discloses/suggests the *link interface* maintains a TLS-secured communication link over the Internet (*maintain[s] a respective secure Internet data message link*) *between* MM1 Transfer Protocol with TCP/IP and TLS on Relay/Server (*link interface*) and a User Agent (*a respective device-link agent*) on each of the UEs/mobile devices (*on each of a plurality of wireless end-user devices*). Traynor, ¶¶548-61.

c. 1[b2]

TS-23.140 discloses/suggests 1[b2]. Traynor, ¶¶562-73.

TS-23.140's MMS supports "transport data specific to applications" on UEs/mobile devices, and application-specific MMs may be transmitted over the network. TS-23.140, 15, 54-55.

POSITAs would have understood/found obvious to include multiple applications to receive application-specific MMs via MMS on each UE/mobile device, which was common by 2009. TS-23.140, 54-55; Traynor, ¶563 (citing Mostafa, 2-4).

In TS-23.140, each application "need[s] to register with the appropriate MMS User Agent or MMS VAS Application" so it is *authorized* to access MMs. TS-23.140, 54-55, 30; Traynor, ¶¶563-68. Once registered, Relay/Server may deliver application-specific MMs using protocols including TLS to the registered application. TS-23.140, 30, 54-55; Traynor, ¶569; §V.A.2.b (1[b1]).

Because application-specific MMs are transmitted to/from User Agents through Relay/Server (§V.A.2.b (1[b1])), POSITAs would have understood MMs are received via User Agent on the particular UE/mobile device. Traynor, ¶¶570-71.

Accordingly, TS-23.140 discloses/suggests each UE/mobile device has multiple applications (*each of the wireless end-user devices comprising multiple software components*) registered (*authorized*) with User Agent (*device link agent*) on UE/mobile device to *receive* application-specific MMs (*messages*) via User Agent. Traynor, ¶¶572-73.

d. 1[c1]

TS-23.140 discloses/suggests 1[c1]. Traynor, ¶¶574-81.

TS-23.140's MMS includes "a collection of MMS-specific network elements" and enables network communication between them (User Agents, Relay/Server, VAS Applications, external server(s)). TS-23.140, 17; Traynor, ¶¶575-76; §§V.A.2.b-V.A.2.c (1[b1]-1[b2]) (Relay/Server receives messages from User Agents/VAS Applications).

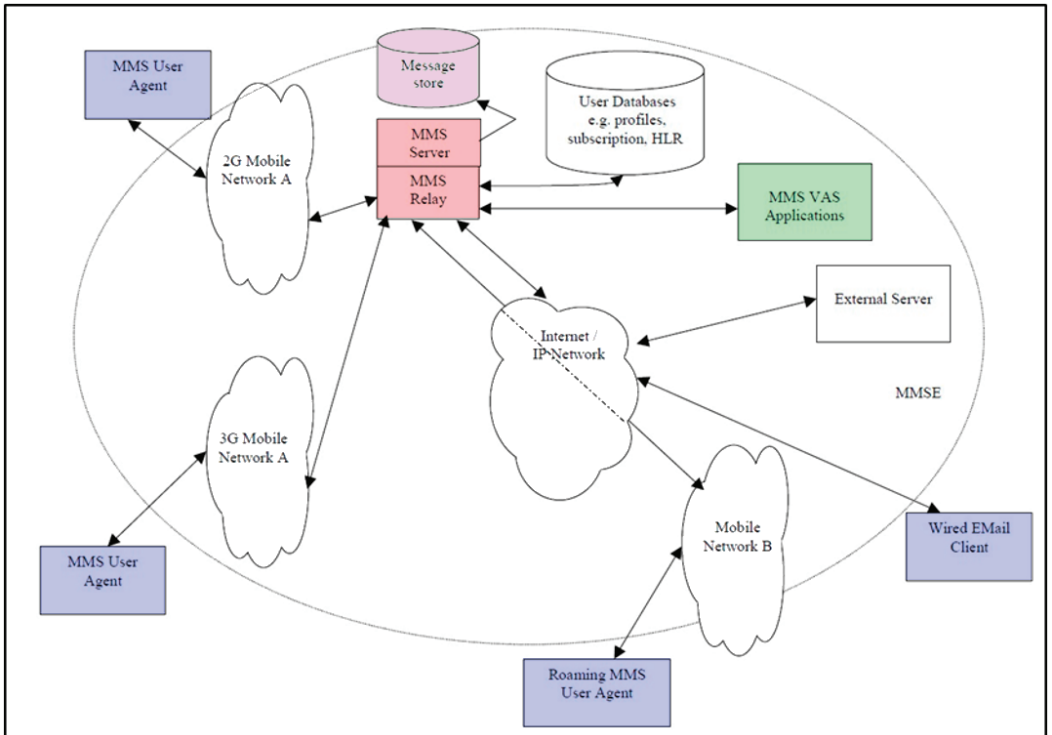
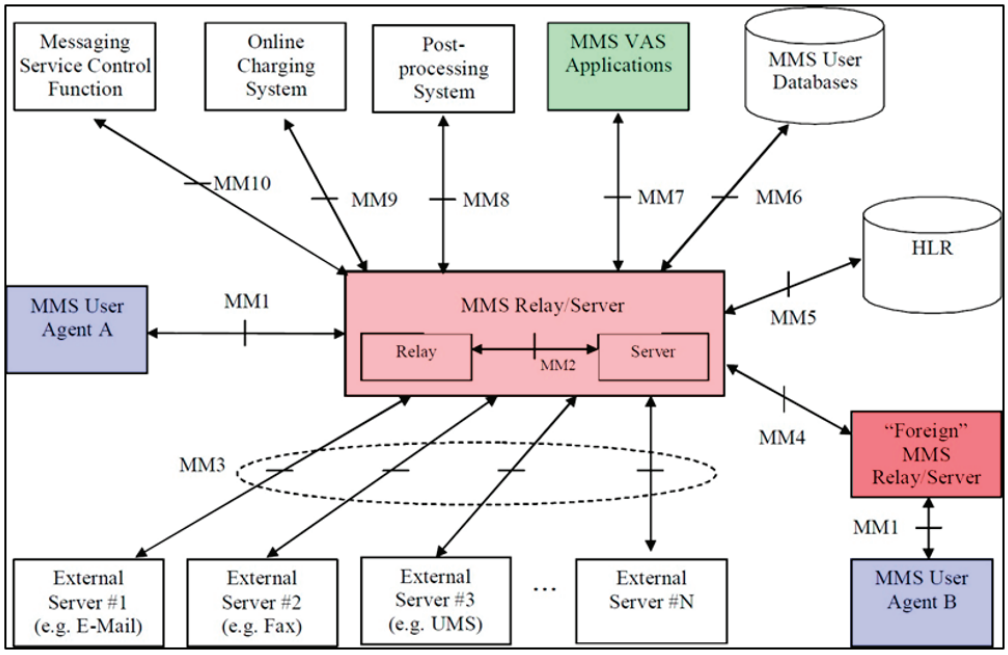


Figure 2: MMS Architectural Elements

TS-23.140, Fig. 2.



TS-23.140, Fig. 3.

MM1 (User Agent-Relay/Server communications)/MM4 (Relay/Server-Relay/Server communications)/MM7 (Relay/Server-VAS Applications communications) are *interfaces* to various networks—including 2G/3G/IP/internet networks (Fig. 2)—and facilitate network communication of messages. TS-23.140, 23-24; Traynor, ¶¶577-79.

Registered applications transmit application-specific MMs to UEs or application servers (on other *network elements*) via Relay/Server. §V.A.2.c (1[b2]); TS-23.140, 54-55; Traynor, ¶¶576, 580. Application-specific MMs include application data and the "application identifier of the destination application." TS-23.140, 54-55. The originator further "indicate[s]" the message recipient's address.

Id., 26, 90, 190 ("Recipient address" in messages over MM1). Relay/Server receives a message and passes "on the destination application identifier" and "application data" to, e.g., another User Agent. *Id.*, 54-56.

Accordingly, TS-23.140 discloses/suggests Relay/Server communicating using MM1/MM4/MM7 protocols over 2G/3G/IP/internet networks (*a network interface*) to *receive messages* from, e.g., User Agents and VAS Applications (*plurality of network elements*) for delivery to registered applications identified by application identifiers in the messages (*for delivery to respective ones of the software components identified in the messages*). Traynor, ¶¶574-81.

e. 1[c2]

TS-23.140 discloses/suggests 1[c2]. Traynor, ¶¶582-89.

TS-23.140's MMS transports application-specific MMs for an application from one device (UE/mobile device, server) and its associated agent (User Agent, VAS Application(s)) to another device and its associated agent. TS-23.140, 54-55; Traynor, ¶585; §§V.A.2.b-V.A.2.c (1[b1]-1[b2]).

Transmission occurs upon an application "trigger[ing]" the User Agent or VAS Application to send an application-specific MM—including application data, "control information," and/or a destination "application identifier"—to a destination application. TS-23.140, 14, 54-56; Traynor, ¶¶583-88.

Recipient/destination applications teach authorized/registered applications (§V.A.2.c (1[b2])) because applications "need to register" using their "application identification value" to utilize MMS. TS-23.140, 54-56; Traynor, ¶¶585-86. Relay/Server receives messages and passes "on the destination application identifier" and "application data" to, e.g., another User Agent. TS-23.140, 54-56; §V.A.2.d (1[c1]); Traynor, ¶¶587-88. These registered applications utilize MM1 Transfer Protocol with TCP/IP and TLS security to communicate. §V.A.2.b (1[b1]); Traynor, ¶¶548-61.

Accordingly, TS-23.140 discloses/suggests each network device has registered applications to send application-specific MMs using, e.g., MM1 Transfer Protocol with TCP/IP and TLS (*each network element authorized to send messages via the link interface*) to registered applications on another device (*to one or more of the software components on one or more of the wireless end-user devices*). Traynor, ¶¶582-589.

f. 1[d1]-[d2]

TS-23.140 discloses/suggests 1[d1]-[d2]. Traynor, ¶¶590-97.

TS-23.140's Relay/Server receives network-element messages with requests to deliver the messages to other application(s) on other UE/wireless devices. Traynor, ¶591; §§V.A.2.d-V.A.2.e (1[c1]-1[c2]). In that process, Relay/Server "stores"

messages and "handl[es]" transfer "between different messaging systems." TS-23.140, 17, 21; Traynor, ¶¶592-93.

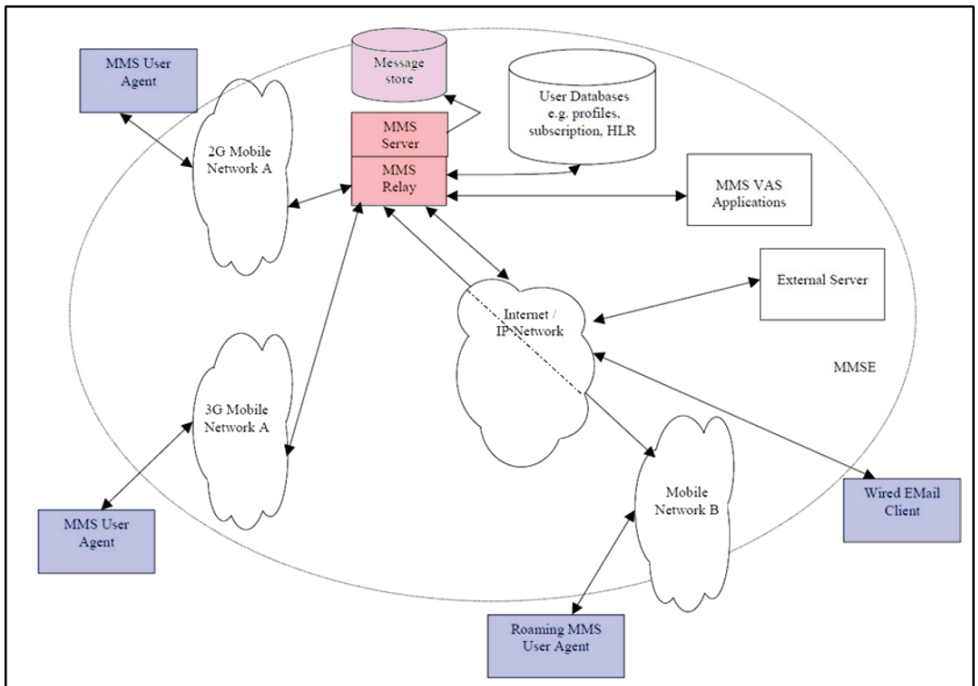


Figure 2: MMS Architectural Elements

TS-23.140, Fig. 2.

Originator Relay/Server "retain[s] the MM until the earliest desired time of delivery," and the recipient Relay/Server (which can be the same as the originator server) "store[s] the MM at least until" "the associated time of expiry is reached, the MM is delivered, [or] the recipient MMS User Agent requests the MM to be routed forward or the MM is rejected." TS-23.140, 26-28. Messages may be stored in Persistent Network-Based Storage" (MMBox) associated with Relay/Server. *Id.*, 21-22, 26-28; Traynor, ¶¶594-95.

Accordingly, TS-23.140 discloses/suggests Relay/Server has a memory (message store and/or MMBox) and logic for delivering messages using triggers (*message-buffer system including a memory and logic*). Traynor, ¶595; §§V.A.2.g-V.A.2.h (1[d3]-[d4]). TS-23.140 discloses/suggests message store and/or MMBox and associated memory stores received messages before delivery to another network element such as a UE/wireless device (*buffer[ing] content from the received network element messages for which delivery is requested to any of the wireless end-user devices*). Traynor, ¶¶594-97.

g. 1[d3]

TS-23.140 discloses/suggests 1[d3]. Traynor, ¶¶598-610.

In TS-23.140, messages are delivered from Relay/Server to User Agent when Relay/Server sends a notification, and User Agent responds with a retrieval request for delivery. TS-23.140, 66-69; Traynor, ¶¶599-600. Several conditions trigger User Agent to issue the retrieval request and secure delivery. Traynor, ¶601. Each of the seven "triggers" below meets the ordinary meaning of that term. Petitioners break them into categories for ease of analysis in case PO argues or the Board finds the ordinary meaning is more limited.

For instance, if a "trigger" must be a condition after which the next step is an attempt to deliver the message, TS-23.140 discloses:

(1) When recipient User Agent is configured for manual retrieval, User Agent manually issues a retrieval request (trigger) in response to a notification to initiate a delivery attempt. TS-23.140, 28-29.

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

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

If a "trigger" encompasses conditions that start/restart an attempted delivery, in addition to at least (1) and (3), the following are also triggers:

(4) When Relay/Server receives a message without a specified delivery time (trigger), it immediately initiates the notification/retrieval process for a delivery attempt. *Id.*, 26-27.

(5) When Relay/Server receives a message with a specified delivery time, the expiration of a timer or event occurrence at the delivery time (trigger) initiates the notification/retrieval process for a delivery attempt. *Id.*

(6) When Relay/Server receives a retrieval request from recipient User Agent with a request for deferred delivery, expiration of a timer or event occurrence at the delivery time (trigger) initiates a delivery attempt. *Id.*, 28-29.

If a "trigger" further encompasses conditions imposed on the messages dictating whether a message will be delivered, in addition to (1)-(6), the following is another trigger:

(7) When Relay/Server receives a retrieval request from recipient User Agent with a "size restriction," determining the message conforms to that restriction (trigger) initiates a delivery attempt. *Id*., 29-30.

Under any reasonable interpretation, because Relay/Server monitors when these events occur/conditions are satisfied, POSITAs would have understood/found obvious Relay/Server includes logic configured to determine when they have occurred for a given UE/mobile device ([*logic to determine*] *when one of a plurality of message delivery triggers for a given one of the wireless end-user devices has occurred*). Traynor, ¶602.

Further, under any reasonable interpretation, POSITAs would have understood/found obvious for at least some received network element messages, message receipt by memory store/MMBox alone is not a trigger (*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). Traynor, ¶603. If a trigger is a condition controlling whether an attempted message delivery is made (conditions (1)-(7)) or that starts/restarts an attempted delivery (conditions (1), (3)-(6)), the receipt of an MM without a specified delivery-time (condition (4)) is a trigger. However, receipt of an MM with a specified delivery-time (*at least some of the received network element messages*) is not itself a trigger; attempted delivery is only triggered by determining the specified delivery-time condition is met (conditions (5) or (6)). Traynor, ¶603. And if a trigger is a condition after which the next step is attempted delivery (conditions (1)-(3)), the receipt of the MM—with/without a specified delivery-time—is not a trigger. Traynor, ¶603. Under this interpretation, none of the received MMs are triggers—thus, *at least some of the received network element messages* are not triggers. Traynor, ¶603.

Further, under any reasonable interpretation, POSITAs would have understood/found obvious *for at least one of the message delivery triggers, the trigger is an occurrence of an asynchronous event with time-critical messaging needs*. Traynor, ¶604. Unless a trigger must be a condition after which the next step is attempted delivery, MM receipt in (4) triggers attempted delivery (i.e., an asynchronous event with time-critical messaging needs). *Id.* And under any reasonable interpretation, manual retrieval in (1) is a trigger because the user manually causes User Agent to request delivery. TS-23.140, 20; Traynor ¶604. Such

manual retrieval is a message delivery trigger that is *an occurrence of an asynchronous event with time-critical messaging needs*—indeed, the '320Pat discloses that a manual "user request" is a trigger. '320Pat, 38:50-63; Traynor, ¶¶604-10.

h. 1[d4]

TS-23.140 discloses/suggests 1[d4]. Traynor, ¶¶611-15.

TS-23.140's Relay/Server delivers stored messages in response to a trigger. Traynor, ¶612; TS-23.140, 28-31; §§V.A.2.f-V.A.2.g (1[d1]-[d3]). POSITAs would have found obvious that Relay/Server includes logic for performing message delivery upon a trigger event. §V.A.2.g (1[d3]); Traynor, ¶612.

TS-23.140 delivers messages via MM1 Transfer Protocol over a TCP/IP and TLS-based link to a recipient User Agent of a particular UE/mobile device. §V.A.2.b (1[b1]); TS-23.140, 24, Fig. 4; Ex-1010, 22; Traynor, ¶613. Although the limitation introduces a new term lacking antecedent basis (*the transport services stack*)—, Petitioners assume the scope includes the stack of secure protocols comprising the *link interface* of 1[b1]. Traynor, ¶614.[9]

Accordingly, TS-23.140 discloses/suggests once Relay/Server detects a trigger (*upon determining that one of the message delivery triggers has occurred for*

[9] Petitioners reserve the right to argue this term is indefinite in other proceedings.

the given one of the wireless end-user devices), Relay/Server includes logic to supply MMs stored in Message Store/MMBox to MM1 Transfer Protocol with TCP/IP and TLS (*the logic further to supply one or messages comprising the buffered content to the transport services stack*). Traynor, ¶¶611-15. Further, MMs are supplied for delivery on the TLS-secured link (*secure message link*) maintained with User Agent (*device-link agent*) on UE/mobile device (*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*). *Id.*

i. 1[e1]

TS-23.140 discloses/suggests 1[e1]. Traynor, ¶¶616-18.

TS-23.140's plurality of UEs/wireless devices (*wireless end-user devices*) each host a User Agent (*device-link agent*) *configured* to perform 1[e2]-1[e4]. §§V.A.2.j-V.A.2.l (1[e2]-1[e4]); Traynor, ¶¶617-18.

j. 1[e2]

TS-23.140 discloses/suggests 1[e2]. Traynor, ¶¶619-21.

TS-23.140 establishes a TCP/IP and TLS-based link between Relay/Server and User Agent over an Internet network—including wirelessly—to transmit/receive MMs. §§V.A.2.b-V.A.2.c (1[b1]-1[b2]); Traynor, ¶620. POSITAs would have understood each User Agent is configured to *maintain the secure* TCP/IP and TLS-based link (*secure Internet data message link*) *over a wireless network to*

the MM1 Transfer Protocol with TCP/IP and TLS on Relay/Server (*link interface*) for at least some amount of time to transmit/receive MMs. Traynor, ¶¶620-21.

k. 1[e3]

TS-23.140 discloses/suggests 1[e3]. Traynor, ¶¶622-26.

TS-23.140's User Agents receive secure messages (*receive secure Internet data messages*) *from* Relay/Server (*network server system*) over a secure link utilizing MM1 Transport Protocol with TCP/IP and TLS (*over the respective secure Internet data message link*). §§V.A.2.b-V.A.2.h (1[b1]-[d4]); Traynor, ¶¶623-24. TS-23.140's User Agents receive messages collected from a variety of other User Agents, VAS Applications, etc. (*collected from multiple…network elements*), and those messages correspond to multiple applications registered with User Agent on each device (*multiple…software components authorized to receive messages via the device-link agent on that respective device*). §§V.A.2.b-V.A.2.e (1[b1]-[c2]); Traynor, ¶623.

When User Agent/VAS Application transmits an application-specific MM, the MM includes an application identifier and message data for the intended application. Traynor, ¶624. Specifically, the MM includes application data, "control information," and a destination "application identifier." TS-23.140, 14, 54-56; Traynor, ¶¶623-24.

Accordingly, POSITAs would have understood application-specific MMs (*at least a first subset of secure Internet data messages*) *contain both* a destination application identifier (*unique identifier for a corresponding one of the software agents*) and application data (*data to be consumed by that software component*) supplied by the originating network element (e.g., User Agent or VAS Application) corresponding to the application (*the data supplied from a respective network element corresponding to that software component*). Traynor, ¶¶625-26.

l. 1[e4]

TS-23.140 discloses/suggests 1[e4]. Traynor, ¶¶627-34.

TS-23.140's Relay/Server transmits application-specific MMs containing application identifiers (*unique identifier*) to User Agents on UEs/mobile devices using MM1 Transfer Protocol with TCP/IP and TLS for a secure link, and the User Agents route them to registered applications ([*software components/applications*] *that are authorized to access messages*) on the UEs/mobile devices (*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*). §§V.A.2.b-V.A.2.e (1[b1]-[c2]); Traynor, ¶¶628-31.

Because application-specific MMs are transmitted to User Agent using a secure link and routed to the registered application (*software component/application*), POSITAs would have understood they are *securely delivered to a software process corresponding to the given software component* to utilize application data transmitted in the MM; otherwise, transmitting application data would be unnecessary if not operated on by the destination application. Traynor, ¶¶632-34.

3. Claim 2

TS-23.140 discloses/suggests claim 2. Traynor, ¶¶635-38.

TS-23.140's Relay/Server evaluates triggers for each message recipient, one of which is reachability of each device (*at least one of the triggers for each given device specific to one or more states of that given device*). §V.A.2.g (1[d3]); Traynor, ¶¶636-38. The '320Pat confirms that operational status teaches device state. '320Pat, 108:19-20, 180:40.

4. Claim 3

TS-23.140 discloses/suggests claim 3. Traynor, ¶¶639-44.

In TS-23.140, originating User Agent can set an earliest desired delivery-time, and Relay/Server stores the message until that time. TS-23.140, 27; Traynor, ¶640. POSITAs would have understood/found obvious specifying delivery time is a

periodic-timer trigger because the timer is triggered upon message receipt and message delivery is triggered at expiration after a particular period. Traynor, ¶641.

Additionally/alternatively, TS-23.140's "periodic polling" involves User Agent retrieving messages from an external server via Relay/Server. TS-23.140, 14, 90-91; Traynor, ¶642. POSITAs would have understood periodic polling is a periodic-timer trigger because polling happens at regularly-repeating intervals, and—when Relay/Server receives the message at polling timer expiration and earliest desired delivery-time y is not specified—the polling timer's expiration starts a process attempting message delivery (*one of the message delivery triggers is the expiration of a periodic timer*). Traynor, ¶643.

5. Claims 5-6

TS-23.140 discloses/suggests claims 5-6. Traynor, ¶¶645-50.

TS-23.140 triggers delivery by having User Agent (*device-link agent*) manually request delivery from Relay/Server over a TLS-secured link (*secure Internet data message link*) in response to a notification from Relay/Server (*receipt of a transmission on the respective secure Internet data message link from the device link agent of the given one of the wireless end-user devices/a request received from the given device link agent*). §§V.A.2.b (1[b1]), V.A.2.f (1[d1]-[d2]), V.A.2.g (1[d3]); TS-23.140, 28-29; Traynor, ¶¶646-50.

6. Claim 7

TS-23.140 discloses/suggests claim 7. Traynor, ¶¶651-54.

TS 23.140's triggers include receipt of a message not specifying an earliest desired delivery-time, receipt of a manual/automatic retrieval request, or receipt of a message indicating User Agent has become available/reachable (*one of the message delivery triggers is the receipt of a particular message from one of the network elements*). §V.A.2.g (1[d3]); Traynor, ¶¶652-54.

7. Claim 10

TS-23.140 discloses/suggests claim 10. Traynor, ¶¶655-658.

TS-23.140 discloses *secure Internet data messages* sent from *network elements*. §V.A.2.k (1[e3]); Traynor, ¶656. At least some MMs (*second subset of secure Internet data messages*) "may be displayed to the end user with or without any pre-notice," and MMs support specified types of media. TS-23.140, 20. POSITAs would have understood this discloses/suggests user messages intended to be displayed on user interfaces of end users' UEs/mobile devices (*user message from a network element…intended for display on a user interface of a given one of the wireless end-user devices*). Traynor, ¶¶657-58.

8. Claim 11

TS-23.140 discloses/suggests claim 11. Traynor, ¶¶659-63.

TS-23.140's MMS network includes MMS User Databases ("User Database(s)") with "elements that…contain user related information such as subscription and configuration" data, user profiles, access controls, server storage space, delivery rules, and UE/wireless device capabilities. TS.23.140, 18, 20. User Agents may interface with User Databases (through Relay/Server) using, e.g., MM1/MM6 Transport Protocols.

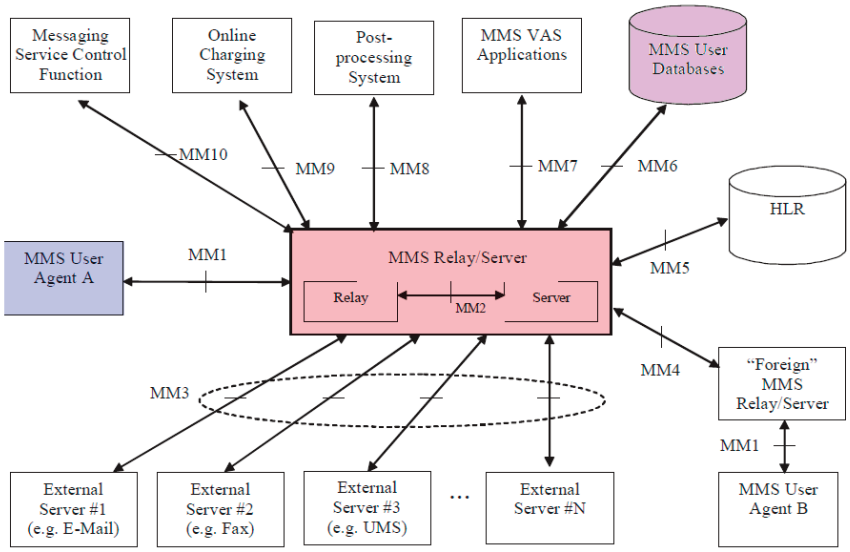


Figure 3: MMS Reference Architecture

TS-23.140, Fig. 3.

POSITAs would have understood the device hosting User Agent has hardware/software element(s) (*policy control agent*) to package for delivery to, or process a delivery from (*data consumed*) User Database, e.g., user-subscription information (*service settings*) or user-defined delivery rules or device capabilities

(*configuration information*) for a device. Traynor ¶¶660-63 (citing Ex-1041, [0032]; Ex-1048, [0003]; Ex-1047, [0037]; Ex-1046, [0032]; Ex-1042, [0043]).

9. Claim 15

a. 15[a]-15[b]

TS-23.140 discloses/suggests 15[a]-15[b]. Traynor, ¶¶664-78.

TS-23.140's User Agent (*device-link agent*) receives an application-specific MM (*upload message*) from an application on UE/mobile device (*software components*) for forwarding to Relay/Server. TS-23.140, 14, 54-55; Traynor, ¶¶665-67. TS-23.140's User Agent can submit/forward to Relay/Server MMs (*upload messages*) with requests to store them. TS-23.140, 40.

TS-23.140 discloses/renders obvious upload application-specific MMs (*upload messages*) from originating User Agent (and associated UE/mobile device) include application data and "application identifier of the destination application." TS-23.140, 54-55; §V.A.2.e (1[c2]). Originating User Agent "indicate[s]" the message recipient's address. TS-23.140, 26, 90, 190; Traynor, ¶¶668-73. Because the message includes a destination application identifier on a particular UE/mobile device and/or indicates the "recipient address," POSITAs would have understood/found obvious each upload message identifies the receiving application and/or UE/mobile device (*each of the upload messages identifying a corresponding*

one of the network elements to which the device respective software component has requested delivery). Traynor, ¶¶668-74.

POSITAs would have recognized routing messages to particular devices is facilitated by including device identification. Traynor, ¶¶675-76. TS-23.140's messages sent to devices include "a user's address, a user's terminal address, or a short code." TS-23.140, 57. Given TS-23.140's disclosures and well-known device-addressing aspects for network communications, POSITAs would have found it obvious to include the recipient address in the upload message for transmitting data between network devices. Traynor, ¶676-78. Accordingly, TS-23.140 discloses/suggests User Agent (*device-link agent*) receives application-specific MMs (*upload messages*) from applications (*software components*) on that device, where the messages include information identifying recipient application and/or recipient UE/mobile device (*each of the upload messages identifying a corresponding one of the network elements to which the device respective software component has requested delivery*). Traynor, ¶¶664-78.

b. 15[c][10]

TS-23.140 discloses/suggests 15[c]. Traynor, ¶¶679-82.

TS-23.140's User Agent (*device-link agent*) transmits *upload messages* to Relay/Server (§V.A.9.a (15[a]-[b])) (*network-message server*), across a link using MM1 Transport Protocol with TCP/IP and TLS (*secure Internet data message link*), (§V.A.2.b (1[b1]), for delivery by Relay/Server (*network-message server*) to identified recipient(s) (*respective identified network elements*) (§§V.A.2.d-V.A.2.e (1[c1]-[c2])). Traynor, ¶¶680-682.

10. Claim 18

TS-23.140 discloses/suggests claim 18. Traynor, ¶¶683-88.

TS-23.140 discloses using MM1 Transfer Protocol with TCP/IP and TLS to send a message with destination application's identifier and application data (*identifier/data pair*). §§V.A.2.b (1[b1]), V.A.2.d-V.A.2.e (1[c1]-[c2]), V.A.2.g-V.A.2.h (1[d3]-[d4]); TS-23.140, 54-56; Traynor, ¶¶684-85.

POSITAs would have understood multiple registered applications on a UE/mobile device receive messages including application data via Relay/Server.

---

[10] Claim 15's *network-message server* lacks antecedent basis. The analysis considers the scope as including claim 1's *network server system*. Traynor, ¶679. Petitioners reserve the right to argue this term is indefinite in other proceedings.

§§V.A.2.d-V.A.2.e (1[c1]-[c2]); TS-23.140, 54-56; Traynor, ¶685 (citing Mostafa, 3-4).

POSITAs would have understood, in some situations, application data/identifiers for multiple applications should be consolidated in a single message. Traynor, ¶686. For instance, message delivery is triggered when User Agent becomes available/reachable (TS-23.140, 28-30), meaning multiple messages may be queued. Traynor, ¶686. Consolidating application data/identifiers for multiple applications into a single message for delivery (*multiple identifier/data pairs*) would provide network efficiencies over sending multiple distinct messages for each application, and would have been readily implemented by combining existing data. Traynor, ¶¶686-87.

Accordingly, TS-23.140 discloses/suggests MMs may comprise identifiers/data from one or more applications (*multiple identifier/data pairs*). Traynor, ¶¶683-88.

B. Ground 1B: TS-23.140-Adamczyk (Claim 3)

Ground 1A explains TS-23.140 discloses/suggests claim 3, which also would have been obvious over TS-23.140-Adamczyk. Traynor, ¶¶689-694.

Like TS-23.140, Adamczyk discloses a message-transmission system. Adamczyk, Abstract, [0007]-[0009]. Adamczyk discloses notification servers (like TS-23.140's Relay/Server) can be "configured to send [] notification messages to

the recipient on demand, at a specific future time, and/or on a periodic schedule" including based on user preferences. Adamczyk, [0010], [0022], cl. 4; Traynor, ¶¶690-91.

POSITAs would have been motivated to supplement TS-23.140's triggers with an additional trigger like Adamczyk's periodic schedule trigger to provide users the option to specify a periodic schedule for message delivery (*one of the message delivery triggers is the expiration of a periodic timer*). Traynor ¶¶691-94. POSITAs would have recognized several benefits from this: (1) providing user control over when messages are received; and (2) increasing UE/mobile device battery life by avoiding repeated pull requests. *Id.*

POSITAs would have had a reasonable expectation of success in doing so because TS-23.140 already taught timers to control message delivery and periodic message-checks. *Id.*, ¶694. To configure Relay/Server to push messages at predetermined, periodic intervals, the combination would have merely required configuring Relay/Server to hold received messages in memory store/MMBox until expiry of a regularly-recurring timer or periodically-scheduled event before pushing all messages received during the time interval at once. *Id.*

C. Ground 1C: TS-23.140-Herzog (With/Without Adamczyk) (Claim 4)

TS-23.140-Herzog or TS-23.140-Adamczyk-Herzog teaches claim 4. Traynor, ¶¶695-700.

TS-23.140 and TS-23.140-Adamczyk teaches claim 3's periodic timer through periodic polling (TS-23.140) and a periodic message delivery schedule (Adamcyzk). §§V.A.4 (cl.3), V.B (cl.3); Traynor, ¶696. But these references do not explicitly disclose mechanisms to keep a connection active/inactive.

Herzog teaches controlling connection duration between client (like TS-23.140's User Agent) and server (like TS-23.140's Relay/Server), e.g., through keep-alive timers to ping a server periodically to keep the connection alive, or to terminate the connection when the keep-alive messages are not sent for a given time period. Herzog, [0036], [0041]-[0043]; Traynor ¶697.

Based on Herzog, POSITAs would have been motivated to implement keep-alive timers and messages in TS-23.140/TS-23.140-Adamcyzk to control connection duration/termination between User Agents and Relay/Server. Traynor, ¶698. Connections would be maintained as long as User Agent indicates its availability and would automatically disconnect when User Agent stops sending keep-alive messages. *Id.* That way, Relay/Server can track connection status of each User Agent and avoid maintaining connections with disconnected/unavailable User Agents. *Id*.

POSITAs would have had a reasonable expectation of success in doing so because TS-23.140 discloses a similar messaging system to Herzog, including a gateway intermediary between server and client. Herzog, [0005], [0024]; TS-23-140, 30. Like TS-23.140, Herzog's messages can be "saved in a message buffer…until [the message] can be delivered to the client" (e.g., when keep-alive messages keep the connection open) (Herzog, [0031]), and using various types of connections, including IP, TCP, SSL and HTTP (*id.*, [0026], [00037]). In other words, the combination merely required implementing keep-alive messaging between User Agent and Relay/Server to maintain TS-23.140's existing connection. Traynor, ¶699.

POSITAs would have understood the periodic polling period (TS-23.140) or periodic-message delivery schedules (TS-23.140-Adamczyk) would be some degree shorter (*fractionally shorter*) than the period required for the keep-alive timers to indicate User Agent availability (*maximum data message interval beyond which the secure message link is taken down*). Traynor, ¶700. If periodic polling/message delivery were set to a longer period than the keep-alive timer period, the connection may be terminated before the message could be sent according to the periodic polling/delivery schedule, thus requiring establishing a new connection. *Id*.

D. Ground 1D: TS-23.140-Shen (Claims 6, 12-13, 16)

1. Shen

Shen addresses security vulnerabilities in MMS (e.g., TS-23.140), which stores messages rather than transmitting them end-to-end. Shen, [0001]-[0002], [0004], [0021]; Traynor, ¶¶108-114. To address this, Shen proposes encrypting the messages using keys/certificates (dark yellow). Shen, Fig. 1, [0021], [0030], [0054]-[0060].

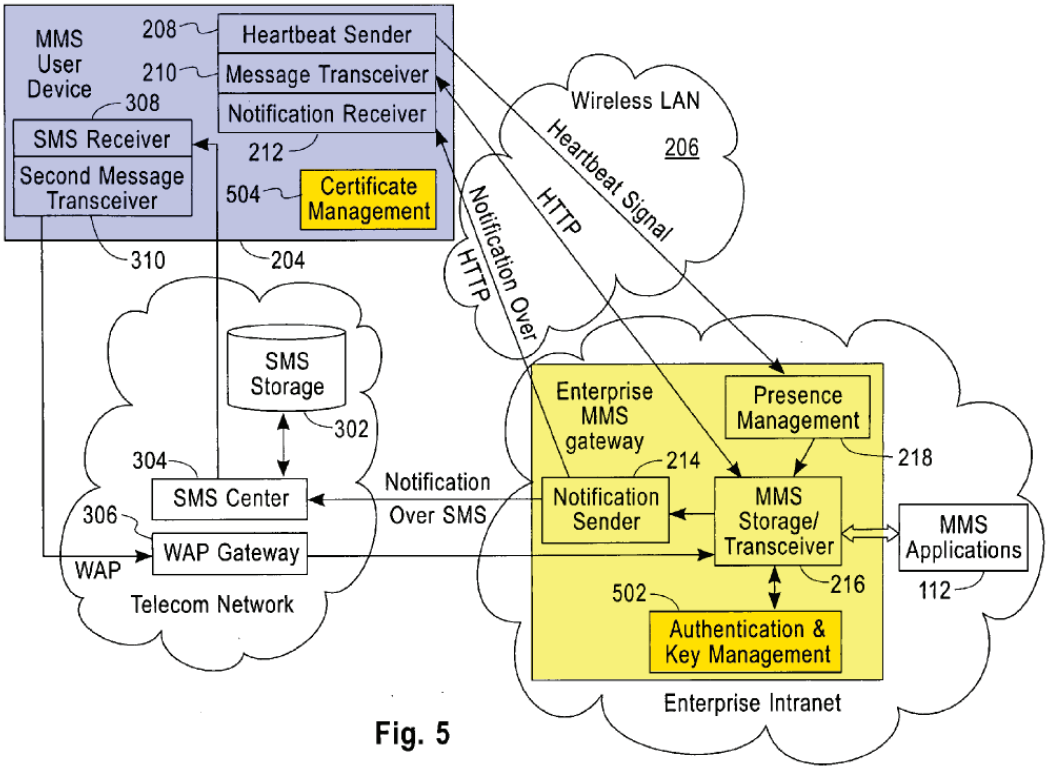


Fig. 5

Shen, Fig. 5.

Shen "sends heartbeat signals periodically" to MMS gateway to determine

LAN availability/MMS user device reachability. *Id.*, [0033]-[0034], [0018]. If gateway receives a heartbeat, the client is available; otherwise, the client is unavailable. *Id.*, [0033]-[0034].

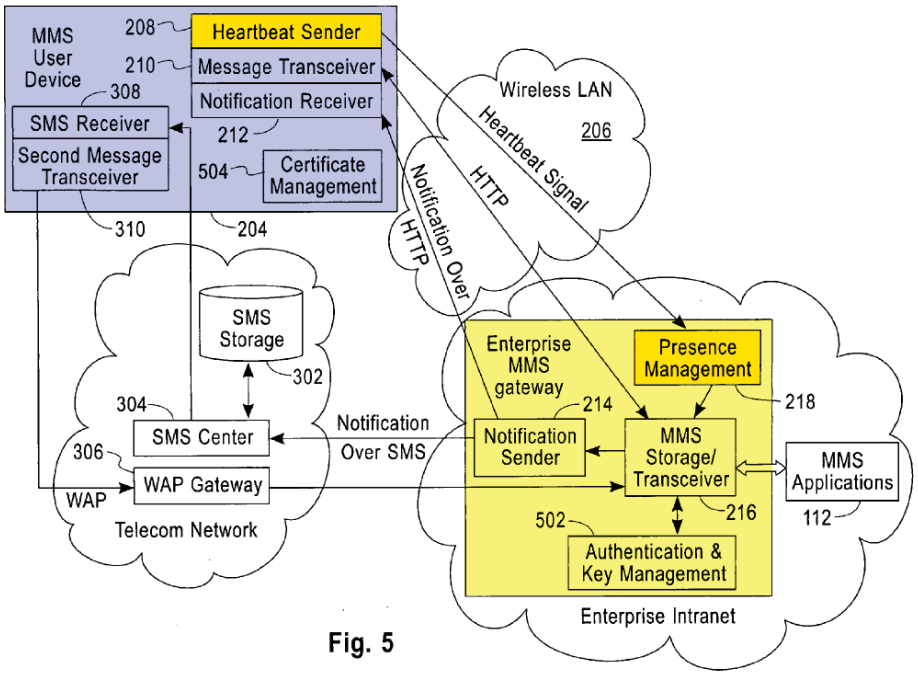


Fig. 5

Shen, Fig. 5

2. Claim 6

TS-23.140-Shen teaches claim 6. Traynor, ¶¶701-06.

Shen's user device generates a *heartbeat message* to indicate device reachability. Shen, [0033]-[0034]; Traynor, ¶¶702-03.

POSITAs would have been motivated to incorporate Shen's "heartbeat" as an additional trigger in TS-23.140. Traynor, ¶704. Specifically, one TS-23.140 trigger occurs when recipient User Agent becomes available/reachable. TS-23.140, 29. Shen's heartbeat was one well-known way to determine whether User Agent has

become "reachable"—i.e., the device pings the gateway (Relay/Server) at intervals to indicate reachability/availability. Shen, [0033]-[0034]; Traynor, ¶¶703-04. POSITAs would have been motivated to implement a *heartbeat message generated by* User Agent (*the given device link agent*) to ping Relay/Server at intervals because this would allow Relay/Server to send buffered messages to UE/mobile device periodically when the device is available/reachable. Traynor, ¶704.

POSITAs would have had a reasonable expectation of success in doing so because it merely would have involved configuring a messaging signal from User Agent to Relay/Server indicating agent's availability/reachability, and Shen already discloses the contents of that message. Shen, [0034]; Traynor, ¶¶705-06.

3. Claim 12

TS-23.140-Shen teaches claim 12. Traynor, ¶¶707-19.

TS-23.140 delivers messages through MM1 Transfer Protocol with TCP/IP and TLS (*link interface*) over a TLS link between Relay/Server and UE/mobile device's User Agent. §V.A.2.b (1[b1]). Although TS-23.140 teaches encrypting the connection, it does not explicitly disclose encrypting messages. Traynor, ¶708.

Shen recognizes failing to encrypt messages in a store-and-forward system like TS-23.140 is a security vulnerability and teaches an authentication/key-management module (*encrypt[ion]*) at Relay/Server (*network server system*) for "encrypting MMS messages." §V.D.1 (Shen); Shen, [0004], [0029]-[0034], [0059],

Fig. 5; Traynor, ¶¶709-10. Shen's messages can be "encode[d] and decode[d] to protect the privacy of these messages." Shen, [0054]; Traynor, ¶711.

POSITAs would have been motivated to modify TS-23.140 to incorporate Shen's encryption/decryption scheme—and specifically add an "authentication and key management module 502" to MM1 Transfer Protocol with TCP/IP and TLS (*link interface*) that generates and "distribute[s] symmetric keys to users" and uses the symmetric keys to encrypt data transmitted to "user device[s]" (Shen, [0055]-[0060])—to improve security by preventing unauthorized access to Relay/Server's stored messages. Traynor, ¶712. This would have involved using a known technique (encryption) to improve similar devices (MMS networks) in the same way (securing content). Further, this would have involved combining prior art elements (encryption/MMS networks) according to known methods (using encryption schemes/key generators) to yield predictable results (secure, encrypted messages). Traynor, ¶¶713-14.

POSITAs would have had a reasonable expectation of success in doing so because Shen envisions its proposal working in coordination with TS-23.140's MMS-system architecture. Shen, [0017]; Traynor, ¶715. Moreover, TS-23.140 and Shen describe similar MMS environments and communications between User Agents and Relay/Server, such that POSITAs would have found it straightforward to modify TS-23.140's Relay/Server to implement Shen's teachings. Traynor, ¶715.

Accordingly, TS-23.140-Shen teaches Relay/Server configured to transport MMs to User Agent on each UE/mobile device via MM1 Transfer Protocol with TCP/IP, TLS, and Shen's encryption scheme (*the link interface to encrypt messages identified for delivery to each given one of the wireless end-user devices to create secure Internet data messages*). Traynor, ¶¶716-19. Further, TS-23.140-Shen teaches User Agent on each UE/mobile device is configured to receive those MMs and apply Shen's decryption scheme before delivering MMs to a registered application/process (*the device link agent on each given device further configured to decrypt the received secure Internet data messages for that device prior to delivering those messages to a respective software process*).

4. Claim 13

TS-23.140-Shen teaches claim 13. Traynor, ¶¶720-22.

TS-23.140-Shen teaches the messages are *encrypted messages*. §V.D.3 (cl.12); Traynor, ¶721. Further, TS-23.140-Shen teaches MMs are transmitted to User Agent using MM1 Transfer Protocol with TCP/IP and TLS (*encryption on a transport services stack*). §V.A.2.b (1[b1])).

5. Claim 16

TS-23.140-Shen teaches claim 16. Traynor, ¶¶723-27.

TS-23.140's terminals may "stor[e] MMs" (*buffer…upload messages*). TS-23.140, 19.

TS-23.140-Shen uses a heartbeat mechanism for the client (TS-23.140's User Agent) "to determine whether a wireless LAN is available." §V.D.2 (cl.6); Traynor, ¶725.

Upon combining TS-23.140 and Shen, POSITAs would have been motivated to configure User Agent (*device-link agent*) to *buffer* messages waiting for transmission until the heartbeat mechanism determined it could reach Relay/Server (*at a time selected by the heartbeat mechanism*). Traynor, ¶726. Such a modification would have, for instance, conserved data because heartbeat messages are generally much smaller in size than MMs, and the system could guarantee Relay/Server's reachability before attempting transmission. Traynor, ¶726 (citing Ex-1043, [0062]; Ex-1044, cl.14). POSITAs would have had a reasonable expectation of success in doing so because it merely involved adjusting timers to trigger message delivery after the heartbeat confirmed Relay/Server's reachability. Traynor, ¶727.

E. Ground 1E: TS-23.140-Pazhyannur (Claims 8-9)

TS-23.140-Pazhyannur teaches claims 8-9. Traynor, ¶¶728-33.

In TS-23.140, applications must register with User Agent to utilize the MMS network (§V.A.2.c(1[b2]), and Relay/Server interacts with User Database that hosts "information for the control of access to the MMS" (TS-23.140, 22; Traynor, ¶729). TS-23.140 does not expressly disclose User Database maintains a list of registered applications/network elements.

Pazhyannur discloses a network of user terminals/application servers, in which a centralized user profile provides a list of authorized devices/applications to permit access to the applications on any authorized user device. Pazhyannur, Abstract, [0040]; Traynor, ¶730.

POSITAs would have been motivated to implement a list of registered network elements and applications (*secure authorization list*) in TS-23.140's User Database (*secure server*) to track applications registered across multiple user devices (Pazhyannur, [0040]), and to "control of access to the MMS" (*indicating the authorized software components and the authorized network elements that are allowed to communicate using the network server system*) (TS-23.140, 22; Traynor, ¶731). Indeed, restricting access to a network using such lists was a well-known means to improve security. Traynor, ¶¶731 (citing Ex-1008, [0044], [0047]). POSITAs would have had a reasonable expectation of success in doing so because TS-23.140 discloses User Database keeps an MMS access control log, and maintaining a centralized list of registered network elements/applications merely required compiling data already held by either User Agents or Relay/Servers (e.g., application and network-element identifiers). Traynor, ¶732.

POSITAs would have appreciated that rather than User Agents or Relay/Server evaluating a local list of registered applications/network elements, they would evaluate centralized list(s) in User Database (*receiving access authorization*

information from the secure server) indicating which applications/network elements are registered to the user and may send MMs to the user's devices (*the access authorization indicating the software components authorized to receive messages via the device link agent on that device*). Traynor, ¶733.

F. Ground 1F: TS-23.140-Ellison (Claim 14)

TS-23.140-Ellison teaches claim 14. Traynor, ¶¶734-53.

TS-23.140 discloses both User Agent (*device-link agent*) and applications executing on UE/mobile device (*given device*) (TS-23.140, 17-18, 23), but TS-23.140 does not expressly disclose executing User Agent in a secure environment. Traynor, ¶735.

Ellison discloses processors are vulnerable to malicious software attacks . Ellison, Abstract, 1:16-51. To inhibit this, Ellison isolates different software elements using a hierarchy of abstract rings and "nubs" to create an "isolated execution mode" (*secure-execution environment*, red) in which hardware/software access "is restricted" compared to a "normal execution mode" (*outside of the secure-execution environment*, green) that "operates in a non-secure/normal environment." Ellison, 4:65-5:1, 6:1-26, 8:25-32, Figs. 1A-1C. Ellison controls access using computer-generated keys. *Id*., 8:66-9:40, 9:47-62, Fig. 2; Traynor, ¶¶736-38.

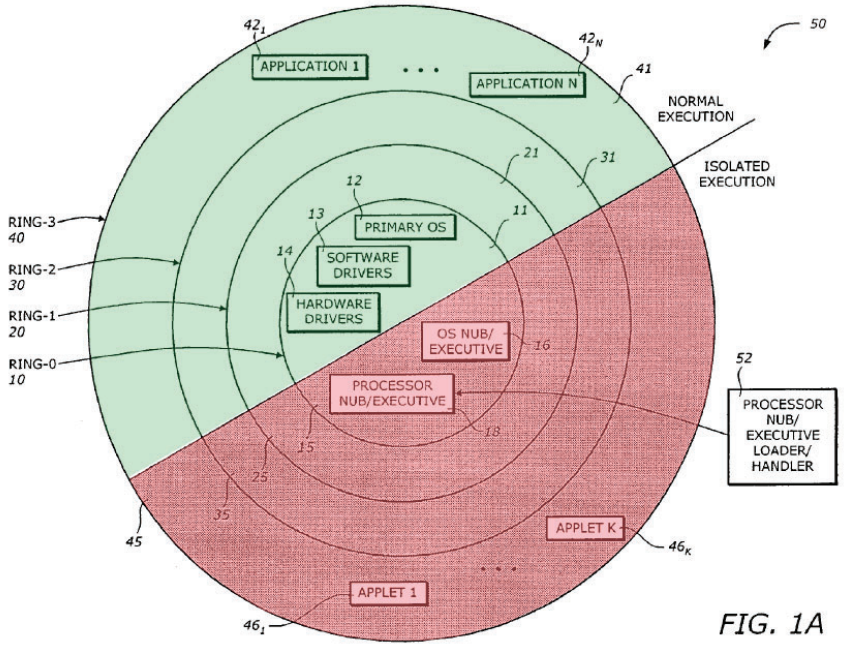


FIG. 1A

Ellison, Fig. 1A.

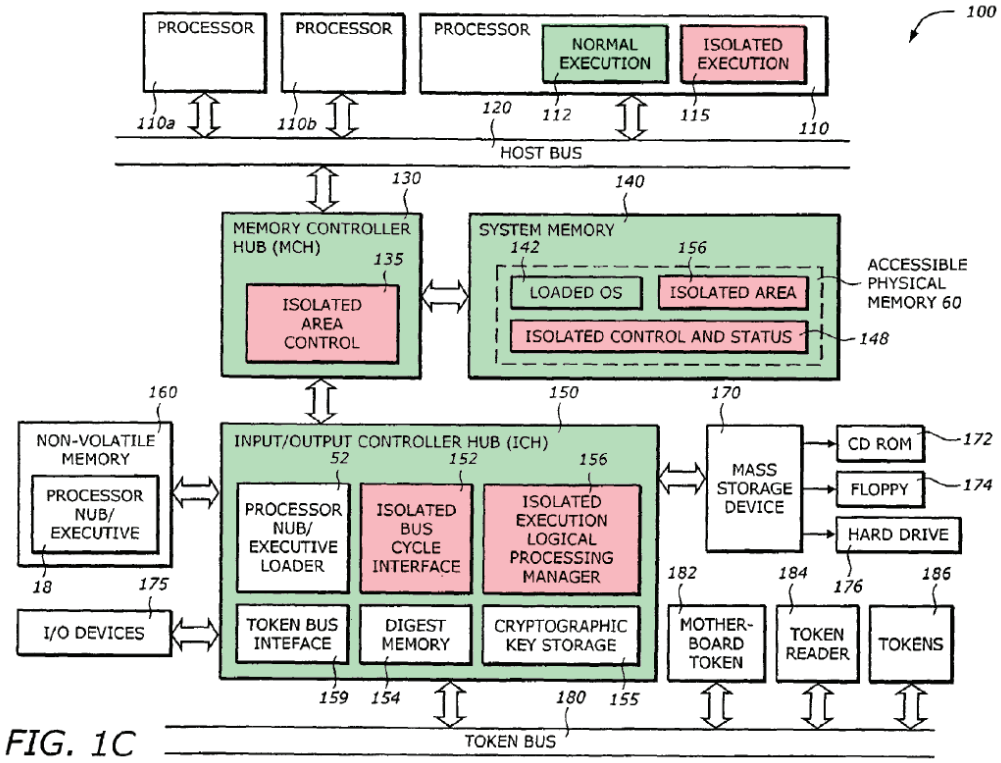


FIG. 1C

Ellison, Fig. 1C.

POSITAs would have been motivated to modify TS-23.140 based on Ellison's teachings of hierarchical, nub-based normal and isolated execution environments to provide enhanced security for User Agent. TS-23.140, 41-42; Ellison, 4:63-5:10, 8:25-32, 8:66-9:6, 9:28-62, Figs. 1A, 2; Traynor, ¶739. Specifically, POSITAs would have understood TS-23.140's User Agent operates in an isolated, secure environment "protected by both the processor and chipset" (Ellison, 3:32-48), to reduce User Agent/user's messages attack exposure. Traynor, ¶¶740-47. Applications interacting with User Agent would operate in normal or separate isolated environments to restrict their access to User Agent. Ellison, 2:55-61.

It was well-known for mobile devices to have (1) messaging services and their associated clients/agents (e.g., push clients) located within the trusted computing environment; and (2) applications residing outside this environment. Traynor, ¶¶745-46 (citing Ex-1033, 3-5). This would have involved combining prior-art elements according to known methods to yield predictable results. Traynor, ¶¶743-50.

POSITAs would have had a reasonable expectation of success in doing so because (1) TS-23.140 contemplates security mechanisms to secure its messaging systems, including messaging around application data; (2) Ellison's techniques would have readily been implemented in "computer system[s]" similar to TS-23.140's network; and (3) it was well-known to use protection rings/tiers in

computing environments (per Ellison). Ellison, 2:46-3:31, 5:11-16, Fig. 1; Traynor, ¶¶748-53 (citing Ex-1033).

Accordingly, TS-23.140-Ellison teaches User Agent (*device-link agent on a given device*) that *executes* in an "isolated" execution environment (*secure-execution environment*), while any applications using MMs through User Agent execute in a "normal" execution environment (*at least one of the software components on that device executes outside of the secure execution environment*).

G. Ground 1G: TS-23.140-Fok (Claim 17)

TS-23.140-Fok teaches claim 17. Traynor, ¶¶754-60.

TS-23.140 transmits application-specific MMs across a network (§V.A.2.c (1[b2]) but does not expressly describe inter-application MM communications (Traynor, ¶755).

Fok describes "secure inter-application communication" in a "mobile operating environment." Fok, [0036]-[0037], [0081], [0106]-[0108], Figs. 4, 8-9. Fok uses a "handshake" between a "primary" and one or more "recipient" applications to establish recipients are trusted using "a list of unique identifiers" before transmitting information between the two. Fok, [0047]-[0053], [0096]-[0102], [0106]-[0108], Figs. 7-10.

POSITAs would have been motivated to modify TS-23.140 based on Fok's teachings to provide "secure inter-application communication" for two or more

applications residing on UE/wireless device to transmit application-specific MMs to one another through User Agent (*at least a given one of the devices further comprising a secure interprocess communication service*). POSITAs would have understood Relay/Server is not involved in the device's inter-application communication (i.e., the communication is *separately secured from the secure Internet data message link*). Traynor, ¶¶756-57. This service would allow trusted applications to transmit application-specific MMs securely to one another on the device via User Agent (*the device link agent for the given device causing messages to be securely delivered to a software process by initiating delivery of each such message on the secure interprocess communication service*). Traynor, ¶¶758-59. POSITAs would have been motivated to provide this because a device may have more than one application installed capable of consuming MM data, and secure inter-application communication would have allowed, e.g., a first application receiving an MM to ensure a second application was trusted before allowing the second application to receive its data. Traynor, ¶759.

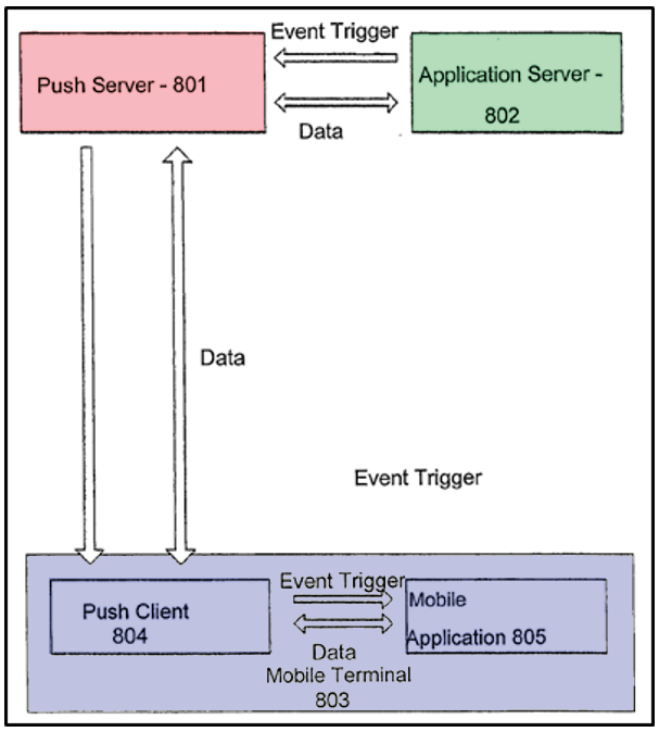
POSITAs would have had a reasonable expectation of success in doing so because Fok describes how to create the inter-application connection. Traynor, ¶760.

H. Ground 2A: Houghton-Munson (Claims 1-7, 10-11, 15, 18)

1. Houghton

Houghton's server "push[es] messages to a" client on a "mobile terminal" in a wireless network. Houghton, Abstract, 11[11]; Traynor, ¶¶122-28. Push messages "may be triggered by any trigger event, local or remote, defined at the server," including an "alarm, notification, or measurement result received to the push server 401 from another device or system." Houghton, 21. Upon triggering, messages are transmitted over a network using secure protocols, e.g., "HTTPS, IP-Sec, secure IP6 or a proprietary security protocol." *Id.*, 19; Traynor, ¶¶123-128.

[11] Citations refer to publication page number.



Houghton, Fig. 8.

2. Munson

Munson discloses a method of "pushing contents to client devices," including "group pushes" in which content is buffered and sent to multiple devices simultaneously, "serializ[ing]" content such that a series of messages are delivered to a particular device simultaneously. Munson, Abstract, [0037], Fig. 4; Traynor, ¶129. The example process (below) shows buffering of received push messages:

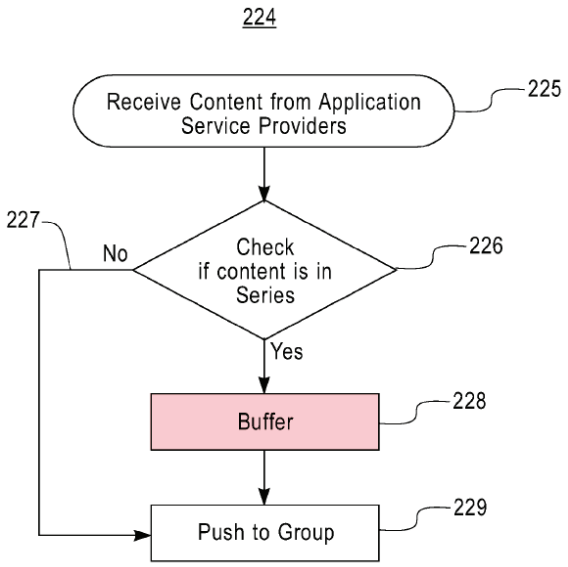


FIG. 4

Munson, Fig. 4.

3. Houghton-Munson Combination

POSITAs would have found it obvious to combine Houghton's and Munson's teachings, and specifically to incorporate Munson's store-and-forward functionality into Houghton's push-messaging system. Traynor, ¶¶761-69.

POSITAs would have been motivated to incorporate computer memory and programming in Houghton to deliver messages from memory, consistent with Munson, to improve system flexibility and reliability. Traynor, ¶762. Providing such store-and-forward functionality would: (1) ensure messages are not lost if a recipient's mobile terminal is unreachable (e.g., if not connected to the network at transmission time) (Traynor, ¶763; Houghton, 3, 7; Ex-1006, [0034], cl.14, Fig. 1);

and (2) provide users greater options to control delivery times/conditions (Traynor, ¶764; Houghton, 14, 21-22, 25-28; Munson, [0040], [0044]). Indeed, combining Houghton-Munson would have amounted to using known prior-art techniques (message buffering) to improve similar devices/systems (push-messaging systems) in the same way to yield predictable results (push-messaging systems that store and forward messages). Traynor, ¶765.

POSITAs would have had a reasonable expectation of success in doing so because Munson provides implementation details (Munson, [0036]-[0038], [0044], Fig. 4) for "store and forward messaging systems" that Houghton already contemplates (Houghton, 3, 7). Traynor, ¶766. The only required changes to Houghton's system would be including generic computer memory at push server to store messages, and adding programming to deliver messages from that memory (including upon occurrence of specific conditions). Traynor, ¶¶766-68.

If Houghton does not expressly disclose specific message-delivery conditions, POSITAs would have been motivated and had a reasonable expectation of success in incorporating conditions Munson teaches to provide additional flexibility in how messages are delivered. *Id.*, ¶769. Houghton discloses conditioning message delivery upon "any trigger event, local or remote." Houghton, 21; Traynor, ¶769. Munson provides additional examples of message delivery triggers (Munson, [0040],

[0044]), which would have required only routine programming within POSITAs' level of skill. Traynor, ¶769.

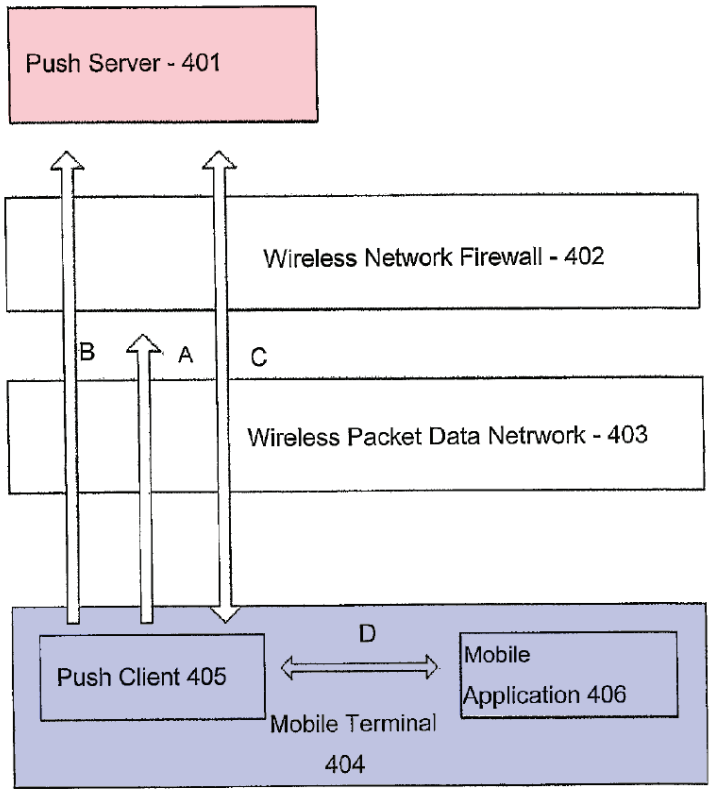
4. Claim 1

a. 1[pre]-1[a]

Houghton discloses/suggests 1[pre] (if limiting)/1[a]. Traynor, ¶¶770-74.

Houghton's network of servers, clients, and computing devices (*networked system*) includes (i) push server (401/801)[12] (*network server system*) to send push messages; and (ii) push clients (405/804) (*device link agents*) on mobile terminals (404/803). Houghton, Abstract, 16-17, 21, Figs. 4, 8; §§V.H.4.b-V.H.4.l ([1b1-e4]); Traynor, ¶¶771-73.

[12] POSITAs would have understood/found obvious to implement features regarding Houghton's Figure 4/6-8 embodiments together. Traynor, ¶771 n.8.

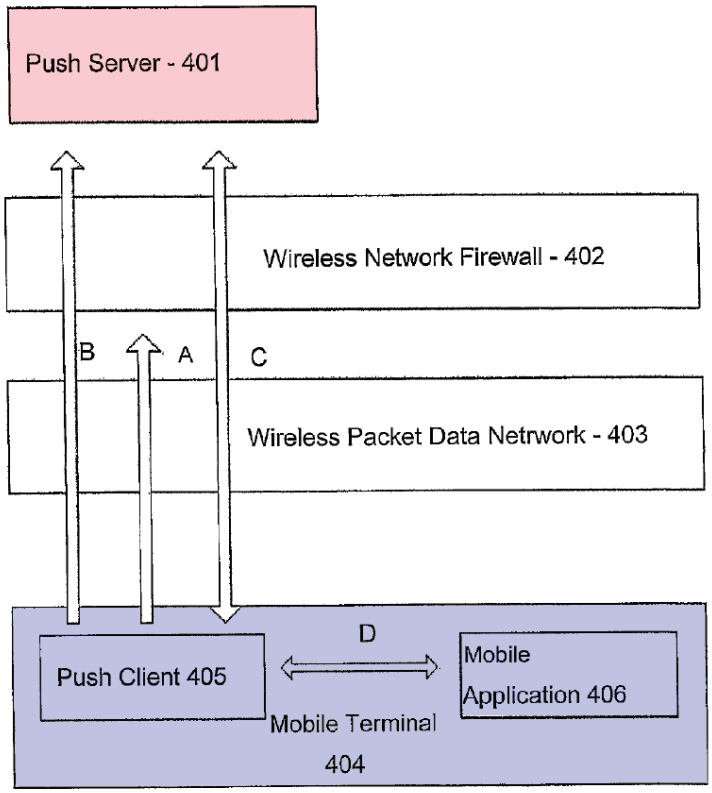


Houghton, Fig. 4.

b. 1[b1]

Houghton discloses/suggests 1[b1]. Traynor, ¶¶775-789.

Push-client software (405) on mobile terminal (404) connects to "push server 401 over a data network 403" (e.g., "a wireless network" for "transmitting Internet Protocol (IP) packets"). Houghton, 17, Fig. 4; Traynor, ¶¶775-78.



Houghton, Fig. 4.

Push client (405/804) teaches a *device-link agent* because it is "software" running on mobile terminal (404) that "initiates and maintains" a link using "Internet technologies" between terminal and push server (401), and using this link, push server (401) pushes messages to mobile terminal (404). Houghton, Abstract, 11, 20; '320Pat, 43:18-50; Traynor, ¶779.

This connection uses known transport protocols, including "connection-oriented protocol[s]" such as "TCP/IP," "connectionless protocol[s]" such as UDP/IP, "alternate connection-oriented protocol[s]" such as HTTP, or "secure protocol[s]" such as "HTTPS, IP-Sec, secure IP6 or a proprietary security protocol"

preventing third-party message interception/modification and identifying communicating parties. Houghton, 18-20, cl.10. Upon connection establishment, push client and push server "push a message" to each other (Figure 4's arrow C). *Id.*, 20, Fig. 4; Traynor, ¶781. Thus, Houghton describes server-client communications using secure-transport protocols (*secure Internet data message link*) like those in the '320Pat. Traynor, ¶¶781-82; '320Pat, 16:66-17:22, 39:20-32, 99:8-32, 100:21-28.

POSITAs would have understood/found obvious Houghton's push server includes secure-transport protocols for establishing a link/transmitting messages over an Internet network with mobile terminal (404). Houghton, 1, 17-20, Fig. 1; Traynor, ¶¶780-82. Indeed, using protocol stacks for securely communicating over a network was well-known and conventional. Houghton, 1; Traynor, ¶¶783-84 (citing Ozaki, [0012]-[0022], Figs. 25-26). POSITAs would have understood Houghton discloses a "stack" of transport protocols. Traynor, ¶¶782, 785.

The communication link established between push client/mobile terminal and push server (401) is *a secure Internet data message link* because it facilitates sending push messages between push client (405) and push server (401) using secure/encryption protocols (e.g., IPsec, HTTPS, SSL) over a network link. Houghton, 18-20, cl.10; Traynor, ¶786; '320Pat, 16:66-17:22, 39:20-32, 70:31-44. POSITAs would have understood/found obvious Houghton's push server's *link*

interface uses such protocols for securing the TCP/IP-based communication link between push client (405/804) and push server (401/801). Traynor, ¶787.

POSITAs would have understood, in push messaging (per Houghton), multiple mobile terminals/associated push clients would communicate with push server (401), and thus form secure SSL or IPSec-based TCP/IP communication links between push server and each respective terminal/push client. Traynor, ¶788; Houghton, 18-19 (multiple "terminals").

Accordingly, Houghton discloses/suggests push server using well-known TCP/IP transport protocols with SSL (*link interface*) *to maintain* a secure SSL or IPSec-based TCP/IP Internet connection (*secure Internet data message link*) *between* TCP/IP transport protocols with SSL on push server (*link interface*) and push client on mobile terminals (*a respective device link agent on each of a plurality of wireless end-user devices*). Traynor, ¶¶775-89.

c. 1[b2]

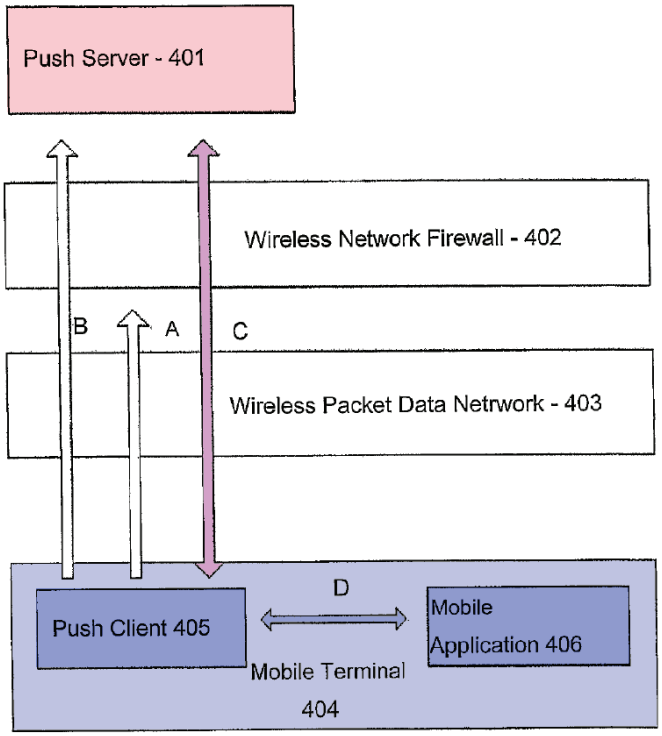
Houghton discloses/suggests 1[b2]. Traynor, ¶¶790-809.

Mobile terminal (404) (*wireless end-user device*) includes a plurality of mobile applications (406) (*multiple software components*). Houghton, 21, 22, 25, 27, Figs. 4, 8; Traynor, ¶¶790-91. Mobile applications (406) are configured to receive message data. Houghton, 21; Traynor, ¶794. Specifically, "application-specific data," e.g., "updates, commands or data messages" are "directed to a push application."

Houghton, cl.1, 11-12; Traynor, ¶¶795-96. The messages include "a data packet" with "information specifying which mobile application 406 from a plurality of such applications" and additional "information to be passed to the…specified mobile application." Houghton, 21-22; Traynor, ¶797.

Push client (405) operating on mobile terminal (404) receives these messages over a secure TCP/IP link before passing them to the relevant application (406). Traynor, ¶793. When push client (405) connects to push server (401), "the server may send a push message" to push client (405) through the "previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS" (Figure 4, arrow C). Houghton, 20, cl.33; Traynor, ¶793; Houghton, 16-17 (push client (405) operates in software); '320Pat, 43:18-31 (device-link agents "implemented largely or entirely in software").

Figure 4 shows communication between push server (401)/push client (405) (communications A/C) and push client (405)/application(s) (406) (communication D). Traynor, ¶791.



Houghton, Fig. 4.

POSITAs would have understood/found obvious push server (401) communicates push messages to multiple "mobile terminals," each including its respective push client (405) and mobile applications (406). Traynor, ¶794; Houghton, 21.

Because push messages are routed to a particular application among multiple applications, and the destination application operates on the received message data, POSITAs would have understood/found obvious these applications are authorized to receive/process data included in command push/application command messages. Traynor, ¶800.

If Houghton does not expressly disclose authorizing applications to receive/process data from secure messages, the disclosed framework suggests it and POSITAs would have considered that obvious. Traynor, ¶¶801-07. Applications were well-known to register with a push server and/or push client before receiving messages via push frameworks. Traynor, ¶¶802-803 (citing Lee, [23]; Shenfield, [0017], [0109]; TS-23.140, 54-56).

POSITAs would have found it obvious to implement well-known authorization/registration processes to enable application registration with a push client and/or push server. Traynor, ¶804. POSITAs would have been motivated to do so for multiple reasons: (1) ensuring application compatibility with push client/server communication protocols; and (2) enabling dynamic content delivery "to have information or data pushed" to devices without device users having to "seek out that data" and ensuring resource-efficient message delivery. Traynor, ¶¶805-06 (citing Shenfield, [0003]-[0006]). Because Houghton contemplates push clients/servers coordinate message delivery to/from mobile applications, implementing well-known application registration/authorization teachings in push environments would have been straightforward and POSITAs would have had a reasonable expectation of success in doing so. Traynor, ¶807.

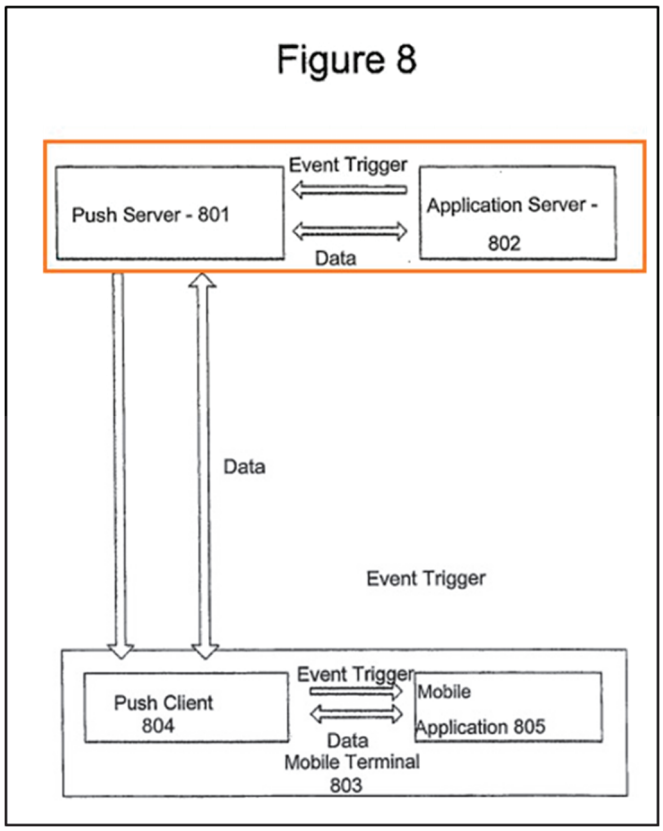
Accordingly, Houghton discloses/suggests mobile terminals (404) (*wireless end-user device*) each comprise multiple applications (406) (*multiple software*

components) *authorized* to receive messages via push client (405) (*device-link agent on that device*). Traynor, ¶¶790-809.

d. 1[c1]

Houghton discloses/suggests 1[c1]. Traynor, ¶¶810-21.

In Houghton's "COMMAND PUSH" procedure, "push server 701 is triggered by a trigger event" from "application server 702 to push an application command message to the push client 704." Houghton, 21-22. A "data connection between application server 802 and mobile application 805 is established" using "[a]n IP or other API…connection between application server 802 and push server 801." *Id.*, 23, 14, 21; Traynor, ¶¶812-15.



Houghton, Fig. 8.

Accordingly, POSITAs would have understood/found obvious application server (802) communicates over a network using API/IP connection with push server (401). Houghton, 23; Traynor, ¶¶812-15. POSITAs would have also understood/found obvious Houghton's push environment includes multiple application servers, each communicating with push server to exchange data/messages with mobile terminal(s). Traynor, ¶¶817-18; Houghton, 21.

The message push server sends to push client includes "a data packet" with "information specifying which mobile application 406 from" multiple "applications" is the message recipient. Houghton, 21, cl.15. POSITAs would have understood/found obvious the message push server's *interface* receives includes information identifying recipient application. §V.H.4.k (1[e3]); Traynor, ¶819 (citing Ex-1006, [0013], [0022] (push message includes application's "app_ID"); Lee, [0011], [0022]-[0023], cl.1).

Accordingly, Houghton discloses/suggests an API (*network interface*) receiving application-specific *messages* from application server(s) (702, 802) (*plurality of network elements*) for delivery to a mobile application "specif[ied]" in the received message (*for delivery to respective ones of the software components identified in the messages*). Traynor, ¶¶810-21.

e. 1[c2]

Houghton discloses/suggests 1[c2]. Traynor, ¶¶822-29.

Push server (401/801) receives push messages from application server (802) and sends them to one of a plurality of mobile applications (*one or more of the software components*) through push client (405/804) on mobile terminal (404/803). Houghton, 21-22; Traynor, ¶822. The message is sent using "a data connection between application server 802 and mobile application 805," using the data connection (Fig. 4, arrow C) between push server 401/801 and push client 405/804. Houghton, 23; Traynor, ¶824.

Because push messages are routed to a particular application among multiple applications, and the destination application operates on the received message data, POSITAs would have understood/found obvious applications are authorized to send, receive, and process data included in command push/application command messages. Traynor, ¶827.

If Houghton does not expressly disclose authorizing mobile terminals, application servers, and their associated applications to send and receive push messages, POSITAs would have considered that obvious. Traynor, ¶¶822-28. Applications on such devices were known to register with push server and/or push client before sending/receiving messages via push frameworks. Traynor, ¶¶802-03,

828 (citing Lee, [0023]; Shenfield, [0017], [0109]; TS-23.140, 54-56); §V.H.4.c. (1[b2]).

POSITAs would have found it obvious to implement well-known authorization/registration processes to enable device and associated application registration with push client and/or push server. Traynor, ¶828. POSITAs would have been motivated to do so for multiple reasons: (1) ensuring device/application is compatible with push client/server's communication protocols; and (2) enabling dynamic content delivery "to have information or data pushed" to devices without device users having to "seek out that data" and ensuring resource-efficient message delivery. Traynor, ¶¶805-06 (citing Shenfield, [0003]-[0006]). Because Houghton contemplates push clients/servers coordinate message delivery to/from mobile terminal applications, implementing well-known application/device registration/authorization teachings in push environments would have been straightforward and POSITAs would have had a reasonable expectation of success in doing so. Traynor, ¶¶807, 828.

Accordingly, Houghton discloses/suggests servers/terminals running applications (*network elements*) *are authorized to send messages via the link interface* to recipient applications (*software components*) on mobile terminals (*end user devices*). Traynor, ¶¶822-29.

f. 1[d1]-[d2]

Houghton-Munson teaches 1[d1]-[d2]. Traynor, ¶¶830-40.

POSITAs would have found it obvious based on Munson to implement a message buffer, including logic, in Houghton's push system to store received push messages on Houghton's push server. §§V.H.4.d-V.H.4.e ([1c1-c2]), V.H.4.g-V.H.4.h ([1d3-d4]); §V.H.3 (motivation); Traynor, ¶¶831-34; Houghton, 3, 7.

Munson stores push messages in push server's memory/storage, including "Series Handler Unit 224" that "receives contents from application service providers" through "Content Receiver Unit 222" and uses a "series buffer…to keep the contents." Munson, [0036]-[0037], Fig. 4. Munson's Figure 3 shows a structure queuing messages in "Content Push Service" (220). Munson, [0035]-[0036]; Traynor, ¶836.

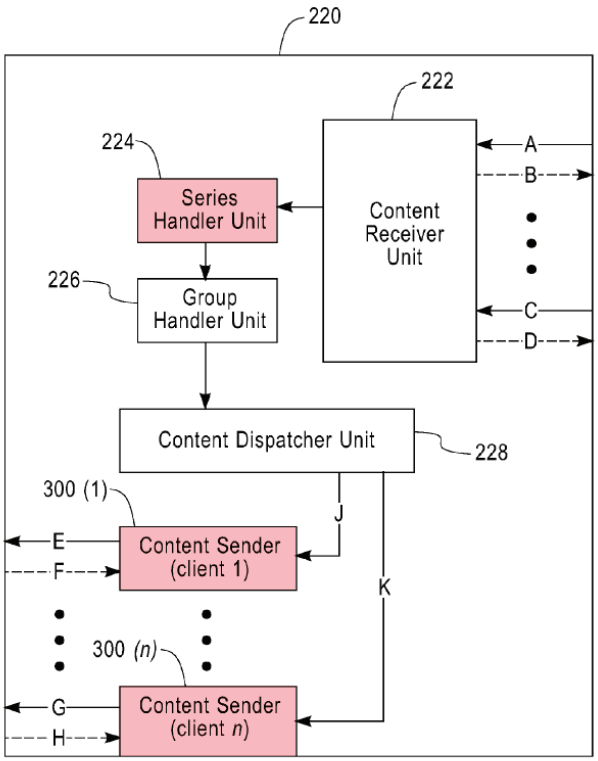


FIG. 3

Munson, Fig. 3.

Multiple "Content Sender[s] 300" include "Queuer 320" and "Push Queues 330." Munson, [0036], [0038], Figs. 3, 5. When pushing messages, Content Senders (300) check the "urgency level of contents," "customer level," and the "dequeue policy," and each "client device" (Houghton-Munson's mobile terminal) is assigned a Content Sender (300). *Id.*; Traynor, ¶837.

Munson's Figure 5 illustrates a structure for queuing messages in Content Sender (300). Traynor, ¶837.

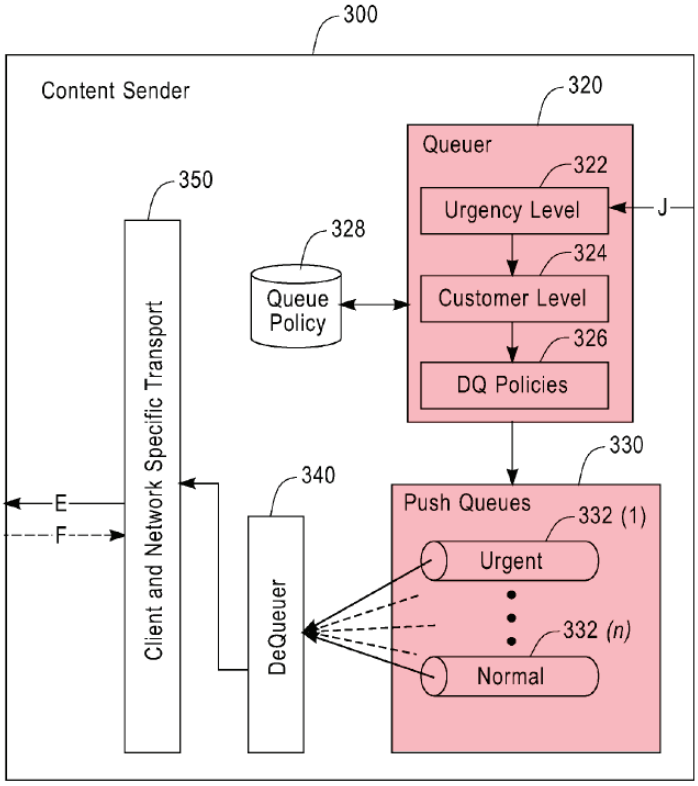


FIG. 5

Munson, Fig. 5.

POSITAs would have found it obvious to implement computer memory and programming to deliver messages from memory, including upon occurrence of certain trigger events. §V.H.3; Traynor, ¶838. Thus, in Houghton-Munson's push server, received network-elements messages are stored in the buffer and associated components. *Id.*

Accordingly, Houghton-Munson teaches a content push service storing messages in push server (*message buffer system*), including a Series Handler Unit (*memory*) and *logic* (addressed below), the memory to *buffer* application-specific

messages for which delivery is requested before transporting them to the client device (*buffer[ing] content from the received network element messages for which delivery is requested to any of the wireless end-user devices*). Traynor, ¶¶830-40.

g. 1[d3]

Houghton-Munson teaches 1[d3]. Traynor, ¶¶841-53.

Houghton's "push message from the push server 401 may be triggered by any trigger event, local or remote, defined at the server," and triggers can "include an alarm, notification, or measurement result received to the push server 401 from another device or system" (*message-delivery triggers [in which] the receipt of a message is not a message delivery trigger*). Houghton, 21, 14; Traynor, ¶¶842-44.

POSITAs would have understood these triggers include an occurrence of *an asynchronous event with time-critical messaging needs*, which includes a user request for message delivery or occurrence of a transaction, consistent with the '320Pat, 38:50-63. §V.A.2.g (Ground 1A, 1[d3]); Traynor, ¶845. Houghton's trigger event can be an alarm or notification received from another device (as described above). Traynor, ¶¶842-43. Message delivery can be adjusted based on whether the message to be delivered is time-critical or "non-time-critical." Houghton, 23-24; Traynor, ¶847.

Houghton's push client also sends a message based on a user request to push server, requesting message delivery. Traynor, ¶849. "IP push command messages"

are triggered when the "push client makes the server aware of" "client-side events," which trigger the push server to deliver messages intended for the terminal's mobile application(s) 406. Houghton, 14, 21; Traynor, ¶849. "[C]lient-side events" include creating "***photographs," "video audio or other media***," "***video game actions or events***," "***messaging actions***" and "***remote application user*** or application server ***event, remote user action***." Houghton, 14, cl.22.

Munson likewise teaches a push system provides "***asynchronous content push*** (i.e., pushing a content) to clients on diverse wireless networks" "according to a schedule of time ***or event***" such as coordinating pushes "for a system maintenance purpose during off-peak hours," thus teaching a message delivery trigger can be *an asynchronous event with time-critical needs*. Munson, [0040], [0044]; Traynor, ¶848. Houghton-Munson teaches this limitation because Houghton contemplates configuring the system for "any trigger event," including those Munson describes as additional example triggers in a similar networking system. Houghton, 21; Traynor, ¶848.

POSITAs would have found it obvious, in view of Houghton/Houghton-Munson, to adjust message delivery based on message time criticality and therefore, adjust message delivery and the associated trigger on an asynchronous basis. Traynor, ¶847.

Houghton-Munson teaches a *buffer* with *logic* to determine when a *trigger* has occurred—including triggers that are not message receipt—including asynchronous content push. Traynor, ¶¶841-53.

h. 1[d4]

Houghton-Munson teaches 1[d4]. Traynor, ¶¶854-58.

Houghton's push server uses a stack of secure protocols for connections between push server/push client (*link interface*; §V.H.4.b (1[b1])) to establish a connection between mobile terminals and push server (*respective secure Internet data message links*; §V.H.4.b (1[b1])). §V.H.4.c (1[b2]); Traynor, ¶¶854-55. Houghton-Munson stores/buffers received push messages in push server's memory, (§V.H.4.f ([1d1-d2]); Traynor, ¶855), and push server includes logic to deliver stored push messages (and associated data/content) to push client upon the occurrence of one or more message delivery triggers (§V.H.4.g (1[d3]); Traynor, ¶856). As noted above, *the transport services stack* lacks antecedent basis and Petitioners assume the scope includes the stack of secure protocols comprising the *link interface* of 1[b1]. §V.A.2.h (Ground 1A, 1[d4])[13]; Traynor, ¶855.

In Houghton, "intelligence" "route[s] messages received on behalf of the wireless terminal to and from the server." Houghton, 14. Such message delivery

[13] Petitioners reserve the right to argue this term is indefinite in other proceedings.

happens using the secure-message link established/maintained between push server and the terminal's push client. Houghton, cl.1; Traynor, ¶857. Upon applying Munson's teachings regarding a memory/buffer (§V.H.4.f ([1d1-d2])), POSITAs would have appreciated Munson's content senders operate similarly to Houghton's routing to transmit messages from the buffer to the client device. §V.H.3 (motivation); Munson, [0036]-[0037]; Traynor, ¶855.

Houghton-Munson teaches push server (*network server*) includes intelligence (*logic*) to *determin[e] that one of the message delivery triggers has occurred*, and in that event send the buffered content (*supply[] one or more messages comprising the buffered content*) to the stack of secure protocols (*transport services stack*) *for delivery on the secure message link maintained between the* stack of secure protocols and the push client (*device-link agent*) on the client device (*wireless end-user device*). Traynor, ¶¶854-58.

i. 1[e1]

Houghton teaches 1[e1]. Traynor, ¶¶859-61.

Houghton's plurality of mobile terminals (*wireless end-user devices*) each have a push client (*device-link agent*) *configured* to perform 1[e2]-1[e4]. Houghton, 21, 23; §§V.H.4.c (1[b2]), V.H.4.j-V.H.4.l (1[e2]-1[e4]); Traynor, ¶860.

j. 1[e2]

Houghton teaches 1[e2]. Traynor, ¶¶862-64.

In Houghton, push client (*device-link agent*) "initiates and maintains…[a] data connection…to the push server using Internet technologies." Houghton, 11, cl.1; §§V.H.4.b ([1b1]), V.H.4.d ([1c1]). Push client connects to push server "over a data network." Houghton, 17, Fig. 4.

Houghton teaches push clients (*device-link agents*) maintaining a secure Internet connection for transmitting/receiving messages with the stack of secure protocols on the push server (*maintain the respective secure Internet data message link over a wireless network to the link interface*). Traynor, ¶¶862-64.

k. 1[e3]

Houghton-Munson teaches 1[e3]. Traynor, ¶¶865-74.

Houghton's push clients receive secure messages (*receive secure Internet data messages*) from push server (*network server system*) over a link using "secure protocol[s]" such as "HTTPS, IP-Sec, secure IP6 or a proprietary security protocol." Houghton, 18-20, §§V.H.4.b-V.H.4.h (1[b1]-[d4]); Traynor, ¶866. Houghton's push clients may receive messages from a variety of other servers and push clients (*collected from multiple…network elements*) and those messages correspond to multiple registered applications (*multiple…software components authorized to receive messages via the device-link agent on that respective device*). §§V.H.4.b-V.H.4.e (1[b1]-[c2]); Traynor, ¶866.

In Houghton, messages sent from application server to application client (through push server) can include: (1) "a data packet containing no information," (2) "information specifying which mobile application 406 from a plurality of such applications" to which the message is directed (*unique identifier for a corresponding one of the software agents*), and/or (3) "information to be passed to the…specified mobile application" (*data to be consumed by that software component*). Houghton, 21; Traynor, ¶867. For example, Houghton discloses "trigger event(s)" or "application command message(s)" are ultimately received by push client (*secure Internet data messages*) and were sent in response to a trigger, which then "trigger[s] [an] event in a mobile application 705 from a plurality of such applications…on the terminal 705" by including "commands or data" (data for the target mobile application) in the message. Houghton, 21-22; Traynor, ¶868. And because these messages are directed to "mobile application 705 from a plurality of such applications" (*id*.), a POSITA would have understood they include an identification of the target mobile application. Traynor, ¶869.

If Houghton does not explicitly disclose messages including a *unique identifier* of the mobile application, POSITAs would have found it obvious to include application identifiers in messages application server sends and push server receives. Traynor, ¶870 (citing Ex-1006, [0013], [0022]). POSITAs would have recognized the application server would be better equipped, relative to the push

server, to provide the application identifier since the application server originates the message and knows the intended application. Traynor, ¶871. Including an application identifier would have amounted to implementing a known technique (including an application identifier in a message) to a known system (Houghton's application server/push server) to achieve predictable results (including the application identifier in the message sent by application server and received by push server). Traynor, ¶872.

POSITAs would have understood application-specific messages received by push client (*at least a first subset of secure Internet data messages*) *contain both* a destination application identifier (*unique identifier for a corresponding one of the software agents*) and application data (*data to be consumed by that software component*) supplied by the originating network element (e.g., application server) corresponding to the application (*the data supplied from a respective network element corresponding to that software component*). Traynor, ¶¶865-74.

l. [1e4]

Houghton-Munson teaches 1[e4]. Traynor, ¶¶875-77.

Houghton's push server transmits application-specific messages containing application identifiers (*unique identifier*) to push clients on mobile terminals using "secure protocol[s]," and push clients route them to registered applications ([*software components/applications*] *that are authorized to access messages*) on

mobile terminals (*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*). §§V.H.4.b-V.H.4.e (1[b1]-[c2]), V.H.4.k (1[e3]); Traynor, ¶875.

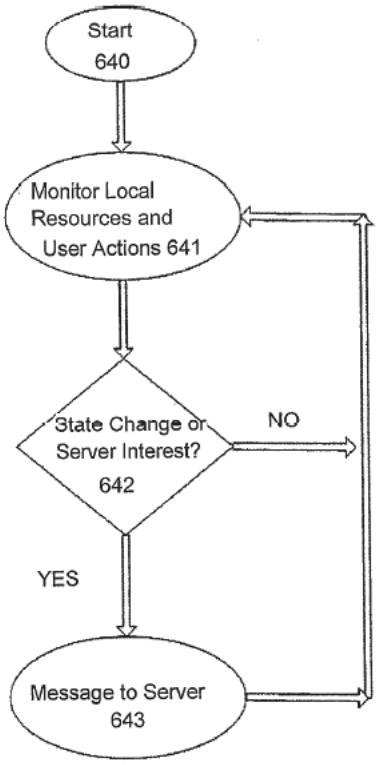
Because application-specific messages are transmitted to push client using a secure link and routed to a registered application (*software component/application*), POSITAs would have understood they are *securely delivered to a software process corresponding to the given software component.* Traynor, ¶875. Houghton's push client can pass "command details (arrow D) to the specified mobile application, module, or additional feature within the same application suite" (*software process[es]*) for processing by the application. Houghton, 21, Fig. 4, cl.1.

5. Claim 2

Houghton-Munson teaches claim 2. Traynor, ¶¶878-80.

In Houghton, logic evaluates a set of trigger conditions, (§§V.H.4.f-V.H.4.h (1[d1]-1[d4])), one of which is "a state change (trigger event)" (*at least one of the triggers for each given device specific to one or more states of that given device*). Houghton, 14, 26, cl.22; Traynor, ¶879.

Figure 6D



Houghton, Fig. 6D.

6. Claim 3

Houghton-Munson teaches claim 3. Traynor, ¶¶881-86.

Houghton contemplates multiple message-delivery triggers (Houghton, 14, 21) and discloses a "periodic message" (sent upon "expiration of a timer") so that devices in the communication path (push client at mobile terminal and push server) "do not time expire the connection," suggesting periodic messages in Houghton recur regularly. *Id.*, 19, 26. "[P]eriodic message timings" can be "measured in seconds, minutes or hours," "longer or shorter…to match the combined needs of all

such network devices and protocols," and "adjusted based on the success or failure of earlier messages," also suggesting these messages recur at regular intervals. *Id.*, 19-20, 26; Traynor, ¶¶882-83.

Munson teaches "content can be pushed according to a schedule of time or event" (*message-delivery trigger*), and provides an example where a "group push" is performed "during off-peak hours" for system maintenance, which POSITAs would have understood/found obvious would be triggered by the expiration of a regularly-repeating timer. Munson, [0044]; Traynor, ¶884; §V.H.4.g (1[d3]).

POSITAs would have been motivated and would have had a reasonable expectation of success in implementing Munson's use of a regularly-repeating timer to cause attempted message delivery in Houghton. Traynor, ¶885. POSITAs would have been motivated to use such a periodic timer as a message delivery trigger to enable more efficient use of network resources, avoid repeated message requests, and attempt delivery according to a desired schedule (*one of the message delivery triggers is the expiration of a periodic timer*). *Id.*

7. Claim 4

Houghton-Munson teaches claim 4. Traynor, ¶¶887-90.

Houghton's push client sends a "periodic message" so the connection does not expire. §V.H.6 (cl. 3); Houghton, 19-20, 26. POSITAs would have understood the period of sending this message sets the *maximum data message interval beyond*

which the secure message link is taken down because if push server does not receive a message at the end of the period, the connection expires. Traynor, ¶888.

POSITAs would have understood/found obvious the period of the regularly-repeating timer for triggering message delivery in Houghton-Munson (*see* §V.H.6 (cl.3)) would be some degree shorter (*fractionally shorter*) than this *maximum data message interval beyond which the secure message link is taken down*. Traynor, ¶889. If the trigger were set to a longer period, the connection would terminate before the message could be sent, thus requiring establishing a new connection. *Id.*

8. Claims 5-6

Houghton-Munson teaches claims 5-6. Traynor, ¶¶891-97.

In Houghton, "client-side events" result in "pushing application commands to a mobile terminal." Houghton, 14, 16, 21, Fig. 4; Traynor, ¶893. Trigger events include "messaging actions" or a "notification." Houghton, 14, 21. Houghton's "client" notifies the "server" of a "terminal event" that triggers a "return message." *Id.*, 20-21. "If a push message is received, the client 405 executes the corresponding function, such as launch or pass commands and data [to] a mobile application." *Id.*, 26.

In Houghton, a connectionless protocol is used "when the client 405 notifies the server 401 of a[] terminal event, user interface event or application event," and push server then "push[es] to the client 405 a service triggered by the event….The

sending of such a return message is frequently time critical and the fastest available combination of techniques will be used." Houghton, 20-21. For established IP connections, push client 405 "monitor[s] the local resources for a state change (a trigger event)" and if so, "send[s] a message to the server 401." Houghton, 26; Traynor, ¶894-95.

An example message trigger includes a periodic message so that devices in the communication path (push client at mobile terminal and push server) "do not time expire the connection," which POSITAs would have recognized/found obvious as a *heartbeat message* generated by the given device-link agent. §V.H.6 (cl.3); Houghton, 26; Traynor, ¶896; '320Pat, 38:50-63.

Houghton-Munson teaches a triggering event for message delivery that is a message from the client to push server, including a heartbeat message (*message delivery trigger is the receipt of a transmission on the respective secure Internet data message link from the device-link agent of the given one of the wireless end-user devices*, including a *heartbeat message* or a *request received from the given device-link agent*). Traynor, ¶¶891-97.

9. Claim 7

Houghton-Munson teaches claim 7. Traynor, ¶¶898-99.

Houghton's "push server 701 is triggered by a trigger event…from an application server 702 to push an application command message to the push client

704 and thereby initiate a mobile terminal client trigger event." Houghton, 21-22, Fig. 7. In other words, Houghton discloses/suggests a message-delivery trigger is the receipt of an application-command message (*particular message*) from an application server (*network element*). Traynor, ¶898.

10. Claim 10

Houghton-Munson teaches claim 10. Traynor, ¶¶900-02.

Houghton discloses *secure Internet data messages* sent from *network elements* (§V.H.4.k (1[e3]); Traynor, ¶901) and at least some messages (*second subset of secure Internet data messages*) include "stimulating…display or shift to [the] foreground of a user interface feature of the terminal such as an application" or a "display notification" (Houghton, 11-12, 18, Fig. 6C). POSITAs would have understood messages displayed to the end user teach a *user message from a network element…intended for display on a user interface of a given one of the wireless end-user devices.* Traynor, ¶901.

11. Claim 11

Houghton-Munson teaches claim 11. Traynor, ¶¶903-05.

Houghton's push client "is also capable of receiving and storing variables which configure behavior of the client," and "may thus specify the network connection preferences, notification sounds, and other behavior of the client." Houghton, 22. POSITAs would have understood Houghton discloses a *software*

component on the client receives and then implements control information (*policy control agent*) including connection preferences and other configurations (*data…compris[ing] service settings and/or configuration information for the device*) from the server. Traynor, ¶904.

12. Claim 15

a. 15[a]-15[b]

Houghton-Munson teaches 15[a]-15[b]. Traynor, ¶¶906-09.

Houghton's push client (405) "accept[s] data D" from multiple mobile applications (406) and directs such data to push server (401) for transmission to "other servers, mobile terminals, push clients and computing devices." Houghton, 21; Traynor, ¶907. The "messages received on behalf of the wireless terminal" may be routed "to and from the server." Houghton, 14. The message's "data packet" can include "information specifying which mobile application 406 from a plurality of such applications" to which it will be directed. *Id.*, 21.

Because Houghton's push client uses a stack of secure protocols for connections between push server and push client (*link interface*; §V.H.4.b (1[b1])), and may encrypt those connections (§V.H.4.b (1[b1])), POSITAs would have understood/found obvious those protocol-based connections *receive* messages (*upload messages*) on push client (*respective device-link agents*) from some of

applications 805 (*software components on the device*) for forwarding to push server. Traynor, ¶908.

POSITAs would have understood/found obvious each such message *identif[ies]*, e.g., the application server (*network element*) *to which* the application issuing the message requested delivery (device respective *software component has requested delivery*). §V.H.4.d (1[c1]); Traynor, ¶909 (citing Lee, [0011], [0022], cl.1).

b. 15[c][14]

Houghton-Munson teaches 15[c]. Traynor, ¶¶910-12.

Houghton's push client (*device-link agent*) transmits the messages (*upload messages*) to push server (*network-message server*) (§V.H.12.a (15[a]-[b])), across a link using a stack of "secure protocol[s]" (*secure Internet data message link*) (§V.H.4.b (1[b1]), for delivery by push server to the identified recipient(s) (*respective identified network elements*) (§§V.H.4.d-V.H.4.e (1[c1]-[c2])). Traynor, ¶911.

13. Claim 18

Houghton-Munson teaches claim 18. Traynor, ¶¶913-18.

[14] *See* n.10; Traynor, ¶910.

Houghton's push messages are received by push server's stack of secure protocols from different network elements such as application server or other push clients resident on a respective mobile terminal, and each message intended for an application executing on one of the mobile terminals includes data and the intended application's corresponding identification. §§V.H.4.c (1[b2]), V.H.4.e (1[c2]); Traynor, ¶915 (citing Lee, [0011], [0022]-[0023], cl.1).

POSITAs would have recognized/found obvious messages received from push server are directed to multiple applications on the mobile terminal (*multiple identifier/data pairs*) because Houghton discloses various "actions" occurring across applications when receiving a push message. Houghton, 21; Traynor, ¶916. The message's "data packet" may include "information specifying which mobile application 406 from a plurality of such applications" it will be directed to. Houghton, 21. In some cases, "packaging of mobile applications involves combining multiple applications delivered in a single bundle" (e.g., a "bundle"/"suite" of applications receive messages packaged together in a message containing *multiple identifier/data pairs*). Houghton, 12; Traynor, ¶916.

I. Ground 2B: Houghton-Munson-TS-23.140 (Claims 1-7, 10-11, 15, 18)

If Houghton-Munson does not disclose/render obvious claims 1-7,10-11, 15, 18—and specifically that applications must register/be authorized or received

message includes an application identifier—it would have been obvious to combine Houghton-Munson with TS-23.140. Traynor, ¶919.

TS-23.140 teaches: (1) applications must register before being granted access to the MMS network (TS-23.140, 54-55); and (2) once registered, application-specific messages include an "application identifier of the destination application," (*id.*, 55). Traynor, ¶920.

POSITAs would have been motivated to incorporate an application-registration process into Houghton-Munson's system to ensure the application is properly configured to transmit messages compatible with the network. TS-23.140, 54; Traynor, ¶921. POSITAs would have had a reasonable expectation of success in doing so because it merely involves negotiated signaling between the registering application and existing messaging-network element (e.g., Houghton-Munson's push client) and TS-23.140 leaves the specific registration process implementation to those in the art to customize. Traynor, ¶921.

Upon registering, POSITAs would have been further motivated to include application identifiers in application messages sent through Houghton-Munson's system, like TS-23.140, to ensure application-specific messages are directed to the intended application or identify an application to another network element. Traynor, ¶922. POSITAs would have had a reasonable expectation of success in doing so because both TS-23.140 and Houghton envision directing messages to specific

applications, and the combination merely involves incorporating an express "application identifier" into the message. *Id*.

If reliance on TS-23.140 is necessary, the combination also teaches all other Challenged Claims as set forth in Grounds 2A/2C-G for the reasons stated therein. Traynor, ¶923.

J. Ground 2C: Houghton-Munson-Adamczyk (Claims 3-4)

Houghton-Munson-Adamcyzk teaches claims 3-4. Traynor, ¶¶924-27.

Adamczyk discloses a message-transmission system. Adamczyk, Abstract, [0007]-[0009]. Adamczyk's notification servers (like Houghton-Munson's push server) can be "configured to send [] notification messages to the recipients on demand, at a specific future time, and/or on a periodic schedule" including based on user preferences. Adamczyk, [0010], [0022], cl.4; Traynor, ¶924.

POSITAs would have found it obvious to supplement Houghton-Munson's triggers (§V.H.4.g (1[d3])) with Adamczyk's additional trigger types. Traynor ¶925. POSITAs would have been motivated to provide users the option to schedule delivery of messages—including on a "periodic schedule," as Adamczyk teaches, sufficient to control when messages are pushed to their devices (*one of the message delivery triggers is the expiration of a periodic timer*). Traynor ¶925. POSITAs would have been motivated to do so with a reasonable expectation of success for the

same reasons described regarding TS-23.140-Adamcyzk. §V.B (Ground 1B); Traynor, ¶926.

POSITAs would have had a reasonable expectation of success doing so because Houghton-Munson already taught timers to control message delivery and periodic messaging. Traynor, ¶926. To configure push server to push messages at periodic intervals, the combination would merely require configuring 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. Traynor, ¶926.

POSITAs would have been further motivated to configure any "periodic schedules" to be *fractionally shorter* than Houghton-Munson's "periodic message" preventing the connection from expiring (*maximum data message interval beyond which the secure message link is taken down*) for the same reasons described regarding Houghton-Munson. §V.H.7 (cl.4, Ground 2A); Traynor, ¶927.

K. Ground 2D: Houghton-Munson-Shen (Claims 6, 12-13, 16)

1. Claim 6

Houghton-Munson-Shen teaches claim 6. Traynor, ¶¶928-32.

Houghton-Munson-Shen teaches a heartbeat message as a trigger. Traynor, ¶928. Houghton uses a "periodic message" as a message-delivery trigger to keep a connection alive, (*see* §V.H.6 (cl. 3)), and monitors for when client (405) has become disconnected (Houghton, 19-20, 26-27; Traynor, ¶929).

If Houghton's "periodic message" is not found to be a heartbeat, POSITAs would have been motivated to implement Houghton's "periodic message" as a *heartbeat* in view of Shen. Traynor, ¶¶930-31. Shen's *heartbeat* was one well-known way to determine whether push client has become "reachable"—i.e., the device pings the server at intervals to indicate reachability/availability. Shen, [0033]-[0034]; Traynor, ¶930. POSITAs would have understood a *heartbeat* provides a low-bandwidth and reliable mechanism for triggering message delivery. Traynor, ¶930.

Implementing Shen's *heartbeat message* would have been a simple substitution of prior art elements (Houghton's periodic message for Shen's heartbeat message) to yield a predictable result (signaling maintenance of a connection and triggering message delivery). Traynor, ¶931. POSITAs would have had a reasonable expectation of success in doing so because Houghton and Shen already provide for similar signaling. Shen, [0034]; Traynor, ¶931.

Houghton-Munson-Shen teaches a message-delivery trigger comprising *a heartbeat message* like Shen's, generated by push client (*device-link agent*). Traynor, ¶¶928-32.

2. Claim 12

Houghton-Munson-Shen teaches claim 12. Traynor, ¶¶933-44.

Houghton supplies messages through secure push-server protocols (e.g., SSL/HTTPS) (*link interface*) for delivery over a secure connection between push

server (*network server system*) and a mobile terminal's push client (*device-link agent on each given device*). §§V.H.4.b (1[b1]), V.H.4.h (1[d4]); Traynor, ¶934. Although Houghton-Munson teaches encrypting the connection, it does not explicitly disclose encrypting the messages. Traynor, ¶934.

Shen recognizes failing to encrypt store-and-forward messages like those in Houghton-Munson presents a security vulnerability, and teaches an authentication/key-management module for "encrypting MMS messages." §V.D.1 (Shen); Shen, [0004], [0029]-[0034], [0059], Fig. 5; Traynor, ¶¶935-37. Shen's messages can be "encode[d] and decode[d] to protect the privacy of these messages." Shen, [0054]; Traynor, ¶936.

POSITAs would have been motivated to modify Houghton-Munson to incorporate Shen's message-encryption/decryption scheme with a reasonable expectation of success for the same reasons described regarding TS-23.140-Shen. §V.D.3 (Ground 1D, cl.12); Traynor, ¶¶936-43.

Houghton-Munson and Shen describe similar messaging environments and communications between push servers and push clients, such that POSITAs would have found straightforward to modify Houghton-Munson's push server to implement Shen's security modules. Traynor, ¶940.

Houghton-Munson-Shen teaches push server configured to transport messages to push client via the TCP/IP transport protocols with SSL and Shen's

encryption scheme (*the link interface to encrypt messages identified for delivery to each given one of the wireless end-user devices to create secure Internet data messages*). Traynor, ¶¶933-44. Houghton-Munson-Shen teaches push client on each mobile terminal configured to receive those messages and apply Shen's decryption scheme before delivering the messages to a registered application/process (*the device link agent on each given device further configured to decrypt the received secure Internet data messages for that device prior to delivering those messages to a respective software process*). *Id.*

3. Claim 13

Houghton-Munson-Shen teaches claim 13. Traynor, ¶¶945-47.

Houghton-Munson-Shen teaches transmitted messages are *encrypted messages* (§V.K.2 (cl.12); Traynor, ¶946), and are transmitted to push clients using "secure protocols" (*encryption on a transport services stack and/or IP layer encryption*) (§V.H.4.b (1[b1]); Traynor, ¶946).

4. Claim 16

Houghton-Munson-Shen teaches claim 16. Traynor, ¶¶948-51.

Houghton tests a connection between push client and push server using periodic messages. §V.H.8 (cl.6). Shen similarly uses a heartbeat mechanism for the client "to determine whether a wireless LAN is available." §V.K.1 (cl.6); Traynor, ¶948.

POSITAs would have been motivated to configure Houghton-Munson's push client (*device-link agent*)—alone or in combination with Shen—to *buffer* messages waiting for transmission until the heartbeat mechanism determined it could reach push server (*at a time selected by the heartbeat mechanism*) for the same reasons as TS-23.140-Shen. §V.D.5 (Ground 1D, cl.16); Traynor, ¶949. POSITAs would have had a reasonable expectation of success in doing so because it merely involved adjusting timers to trigger message delivery after the heartbeat confirmed push server's reachability. Traynor, ¶950.

L. Ground 2E: Houghton-Munson-Pazhyannur (Claims 8-9)

Houghton-Munson-Pazhyannur teaches claims 8-9. Traynor, ¶¶952-58.

POSITAs would have understood applications in Houghton-Munson's system register with push client to transmit/receive messages. §V.H.4.c (1[b2]); Traynor, ¶953. Houghton-Munson does not expressly disclose maintaining a list of registered applications on a server.

Pazhyannur discloses a network of user terminals/application servers, in which a centralized user profile provides a list of authorized devices/applications to permit access to the applications on any authorized user device. Pazhyannur, Abstract, [0040]; Traynor, ¶954.

POSITAs would have been motivated to implement a list of registered network elements/applications (*secure authorization list*) in Houghton-Munson's

server (*secure server*), such as a portion of push server, to track all devices/applications associated with a user (Pazhyannur, [0040]), as well as use such a list to control access to a network to improve network security (Traynor, ¶955; Ex-1008, [0044], [0047]). POSITAs would have had a reasonable expectation of success in doing so because maintaining a centralized list of registered network elements/applications merely required compiling network-element/application-registration data Houghton-Munson's system already has (e.g., application/network-element identifiers). Traynor, ¶956.

POSITAs would have appreciated that rather than push client or push server evaluating a local list of registered applications/network elements, they would evaluate centralized list(s) on a server (such as push server) (*receiving access authorization information from the secure server*) indicating which applications/network elements are registered to the user and may send messages to the user's devices (*the access authorization indicating the software components authorized to receive messages via the device link agent on that device*). Traynor, ¶¶952-58.

M. Ground 2F: Houghton-Munson-Ellison (Claim 14)

Houghton-Munson-Ellison teaches claim 14. Traynor, ¶¶959-76.

Houghton-Munson teaches push client (*device-link agent*) and other applications executing on a mobile terminal (Houghton, 16, 21, Fig. 4), but does not explicitly teach implementing push client in a secure environment. Traynor, ¶960.

Ellison implements software in a secure environment to protect it from potential attacks. §V.F; Traynor, ¶¶961-63.

POSITAs would have been motivated to combine Houghton-Munson with Ellison with a reasonable expectation of success to incorporate Ellison's hierarchical, nub-based normal and isolated execution environments to provide enhanced security for push server for the same reasons described regarding TS-23.140-Ellison. §V.F (Ground 1F); Traynor, ¶964.

POSITAs would have had a reasonable expectation of success in doing so because (1) Houghton contemplates security mechanisms to secure its messaging systems, including messaging around application data; (2) Ellison's techniques would have readily been implemented in "computer system[s]" similar to Houghton-Munson's network; and (3) it was well-known to use protection rings/tiers in computing environments (per Ellison). Ellison, 2:46-3:31, Fig. 1A; Traynor, ¶¶968-73 (citing Ex-1033).

Houghton-Munson-Ellison teaches providing for a push client (*device-link agent*) to execute in an "isolated" execution environment (*secure-execution environment*) like Ellison's, while any applications utilizing messages through push

client execute in either a separate "isolated" environment or "normal" execution environment (*at least one of the software components on that device executes outside of the secure execution environment*). Traynor, ¶¶959-76.

N. Ground 2G: Houghton-Munson-Fok (Claim 17)

Houghton-Munson-Fok teaches claim 17. Traynor, ¶¶977-83.

Houghton transmits application-specific messages across a network, (§V.H.4.c (1[b2]), but does not expressly describe inter-application message communications (Traynor, ¶978).

Fok describes a "secure inter-application communication" in a "mobile operating environment." Fok, [0036]-[0037], [0081], [0106]-[0108], Figs. 4, 8-9. Fok uses a "handshake" between a "primary" and one or more "recipient" applications to establish recipients are trusted using "a list of unique identifiers" before transmitting information between the two. Fok, [0047]-[0053], [0096]-[0102], [0106]-[0108], Figs. 7-10; Traynor, ¶979.

POSITAs would have been motivated to combine Houghton-Munson and Fok to provide "secure inter-application communication" (*secure interprocess communication service)* for two or more applications residing on a single terminal to transmit application-specific messages to one another through push client on the terminal (*at least a given one of the devices further comprising a secure interprocess communication service*). POSITAs would have understood push server is not

involved in the device's inter-application communication (i.e., the communication is *separately secured from the secure Internet data message link*). Traynor, ¶980. This service would allow trusted applications to transmit application-specific messages securely to one another on the terminal via push client (*the device link agent for the given device causing messages to be securely delivered to a software process by initiating delivery of each such message on the secure interprocess communication service*). Traynor, ¶981. POSITAs would have been motivated to combine Houghton-Munson with Fok with a reasonable expectation of success for the same reasons discussed regarding TS-23.140-Fok. §V.G (Ground 1G); Traynor, ¶¶981.

VI. CONCLUSION

IPR of the Challenged Claims is respectfully requested.

VII. MANDATORY NOTICES

A. Real Party in Interest

Petitioners are the real parties-in-interest, along with Amazon.com, Inc. and Apple Inc. 37 C.F.R. § 42.8(b)(1).

B. Related Matters

To the best of Petitioners' knowledge, the '320Pat has been involved in the following matters:

- *Headwater Research LLC v. Amazon.com Services LLC et al.*, No. 7-25-cv-00286 (WDTX).
- *Headwater Research LLC v. Walmart Inc.*, No. 2-25-cv-00961 (EDTX).
- *Headwater Research LLC v. Uber Techs., Inc. et al.*, No. 2-25-cv-00962 (EDTX).
- *Headwater Research LLC v. Target Corp.*, No. 2-25-cv-00963 (EDTX).
- *Headwater Research LLC v. Supercell Oy*, No. 2-25-cv-00964 (EDTX).
- *Headwater Research LLC v. Tencent Holdings Ltd.*, No. 2-25-cv-00965 (EDTX).
- *Headwater Research LLC v. Apple Inc.*, No. 7-25-cv-00371 (WDTX).
- *Headwater Research LLC v. Google LLC*, No. 7-25-cv-00231 (WDTX).

C. Notice of Counsel and Service Information

LEAD COUNSEL
Jessica Kaiser (Reg. No. 58,937) kaiser-ptab@perkinscoie.com PERKINS COIE LLP 1900 Sixteenth Street, Suite 1400 Denver, CO 80202-5255 Telephone: (303) 291-2300
BACK-UP COUNSEL
Christopher Marando (Reg. No. 67,898) Marando-ptab@perkinscoie.com

PERKINS COIE LLP
700 Thirteenth Street N.W.
Suite 800
Washington, DC 20005-3960
Phone: (202) 654-6200

Thomas Millikan (Reg. No. 72,316)
Millikan-ptab@perkinscoie.com
PERKINS COIE LLP
11452 El Camino Real
Suite 300
San Diego, CA 92130-2080
Phone: (858) 720-5723

Matthew A. Lembo (Reg. No. 75,633)
Lembo-ptab@perkinscoie.com
PERKINS COIE LLP
1155 Avenue of the Americas 22nd Floor
New York, NY 10036-2711
Phone: (332) 238-2757

Petitioner consents to electronic service. All services and communications to the attorneys listed above may be sent to:

Amazon-Headwater-IPR@perkinscoie.com

D. Power of Attorney

A power of attorney is filed herewith according to 37 C.F.R. §42.10(b).

Respectfully submitted,

/ *Jessica Kaiser* /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255

Date: November 10, 2025

CERTIFICATE OF WORD COUNT UNDER 37 CFR §42.24(D)

Pursuant to 37 C.F.R. §42.24(a), Petitioner hereby certifies that portions of the above-captioned Petition for *inter partes* review of U.S. Patent No. 10,321,320, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,993. Pursuant to 37 C.F.R. §42.24(a), this word count is in compliance and excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing. This word count was prepared using Microsoft Word.

Respectfully submitted,

/ *Jessica Kaiser* /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

Date: November 10, 2025
PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255

CERTIFICATE OF SERVICE

The undersigned hereby certifies that true copies of the Petition for *inter partes* review of U.S. Patent No. 10,321,320 and supporting materials (Exhibits and Power of Attorney) were served via overnight delivery on the Patent Owner at the correspondence address of record as listed on PAIR:

Headwater Research LLC
C/O Farjami & Farjami LLP
26522 La Alameda Ave., Suite 360
Mission Viejo, CA 92691

A courtesy copy was also sent via electronic mail to Patent Owner's litigation counsel listed below:

Brian D. Ledahl - bledahl@raklaw.com
Dale Chang - dale.chang@lw.com
James N. Pickens - jpickens@raklaw.com
James S. Tsuei - jtsuei@raklaw.com
Jason M Wietholter - jwietholter@raklaw.com
Kristopher R. Davis - kdavis@raklaw.com
Paul A. Kroeger - pkroeger@raklaw.com
Qi (Peter) Tong - ptong@raklaw.com
Reza Mirzaie - rmirzaie@raklaw.com
Marc A. Fenster - mafenster@raklaw.com

Respectfully submitted,

/ *Jessica Kaiser* /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

Date: November 10, 2025