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.

IPR2026-00088

U.S. Patent No. 9,615,192

PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 9,615,192

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

Google Exhibit 1068
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 2
II. '192 PATENT 4
III. NO PRIOR IPR OR DISTRICT COURT HAS RULED ON INVALIDITY OF THE '192PAT 5
IV. LEVEL OF ORDINARY SKILL IN THE ART 5
V. CLAIM CONSTRUCTION 6
VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE 6
A. Ground 1A: TS-23.140 (Claims 1, 5-7, 9, 11-13, 15) 6
1. TS-23.140 6
2. Claims 1, 15 7
3. Claim 5 23
4. Claim 6 27
5. Claim 7 28
6. Claim 9 28
7. Claims 11-12 29
8. Claim 13 30
B. Ground 1B: TS-23.140-Shen (Claims 2-3, 12) 30
1. Shen 30
2. Claim 2 32
3. Claim 3 34
4. Claim 12 34
C. Ground 1C: TS-23.140-Ellison (Claim 4) 35
D. Ground 1D: TS-23.140-Rakic (Claim 8) 38

E. Ground 1E: TS-23.140-Adamczyk (Claim 9) 41
F. Ground 1F: TS-23.140-Herzog (With/ Without Adamczyk) (Claim 10) 42
G. Ground 1G: TS-23.140-Gellens (Claim 14) 44
H. Ground 2A: Houghton-Munson (Claims 1, 5-7, 9-13, 15) 45
1. Houghton 45
2. Munson 46
3. Houghton-Munson Combination 47
4. Claims 1, 15 49
5. Claim 5 68
6. Claim 6 70
7. Claim 7 71
8. Claim 9 72
9. Claim 10 73
10. Claims 11-12 73
11. Claim 13 75
I. Ground 2B: Houghton-Munson-TS-23.140 (Claims 1, 5-7, 9-13, 15) 75
J. Ground 2C: Houghton-Munson-Shen (Claims 2, 3, 12) 77
1. Claim 2 77
2. Claim 3 78
3. Claim 12 79
K. Ground 2D: Houghton-Munson-Ellison (Claim 4) 80
L. Ground 2E: Houghton-Munson-Rakic (Claim 8) 81
M. Ground 2F: Houghton-Munson-Adamczyk (Claims 9-10) 83
1. Claim 9 83
2. Claim 10 84
N. Ground 2G: Houghton-Munson-Gellens (Claim 14) 84
VII. CONCLUSION 85

VIII. MANDATORY NOTICES 86

A. Real Party in Interest 86

B. Related Matters 86

C. Notice of Counsel and Service Information 87

D. Power of Attorney 88

EXHIBIT LIST

No.	Exhibit
1001	U.S. Patent No. 9,615,192 ("'192 patent" or "'192Pat")
1002	File History of the '192 patent ("'192FH")
1003	Declaration and Curriculum Vitae of Dr. Patrick 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. 2009/0158397 to Herzog et al. ("Herzog")
1009	U.S. Patent No. 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. 7,509,487 to Lu et al. ("Lu")
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*

No.	Exhibit
	https://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_19/Docs/PDF/SP-030167.pdf
1014	U.S. Patent Pub. No. 2005/0207379 to Shen et al. ("Shen")
1015	U.S. Patent Pub. No. 2009/0282256 to Rakic et al. ("Rakic")
1016	Declaration of Friedhelm Rodermund
1017	U.S. Patent Pub. No. 2009/0240807 to Munson et al. ("Munson")
1018	EP Patent Application EP1853044 to Shenfield ("Shenfield")
1019	U.S. Patent No. 7,082,615 to Ellison et al. ("Ellison")
1020	U.S. Patent Pub. No. 2005/0144294 to Gellens et al. ("Gellens")
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)
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

No.	Exhibit
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*, 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	RESERVED
1038	RESERVED
1039	U.S. Patent Pub. No. 2003/0126282 to Sarkar et al. ("Sarkar")
1040	U.S. Patent Pub. No. 2007/0037610 to Logan ("Logan")

No.	Exhibit
1041	RESERVED
1042	RESERVED
1043	U.S. Patent Pub. No. 2004/0085894 to Wang et al. ("Wang")
1044	RESERVED
1045	Simon Higginson, *Platform Security Concepts, in* SYMBIAN OS PLATFORM SECURITY: SOFTWARE DEVELOPMENT USING THE SYMBIAN OS SECURITY ARCHITECTURE, 17-41 (Craig Heath ed., 2006) ("Higginson")

LISTING OF CHALLENGED CLAIMS

Reference	Limitation
Claim 1	
1[pre]	A message link server comprising:
1[a]	a transport services stack to maintain a respective secure message link through an Internet network between the message link server and a respective device link agent on each of a plurality of wireless end-user devices,
1[b]	each of the wireless end-user devices comprising multiple software components authorized to receive and process data from secure message link messages received via a device link agent on that device;
1[c1]	an interface to a network to receive network element messages from a plurality of network elements,
1[c2]	the received network element messages comprising respective message content and requests for delivery of the respective message content to respective wireless end-user devices, the respective message content including data for, and an identification of, a respective one of the authorized software components; 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 a given one of the wireless end-user devices,
1[d3]	the logic to determine when one of a plurality of message delivery triggers for the 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

Reference	Limitation
1[d4]	upon determining that one of the message delivery triggers has occurred, 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.
Claim 2	
2	The message link server of claim 1, further comprising an encrypt function to encrypt the one or more messages supplied to the transport services stack for delivery on the secure message link maintained between the message link server and the device link agent on the given one of the wireless end-user devices.
Claim 3	
3	The message link server of claim 2, wherein the encrypted one or more messages are transported to the device link agent on the given one of the wireless end-user devices using one or more of encryption on the transport services stack, IP (Internet Protocol) layer encryption, and tunneling.
Claim 4	
4	The message link server of claim 1, wherein the device link agent executes in a secure execution environment on at least one of the devices, and at least one of the software components executes outside of the secure execution environment on that device.
Claim 5	
5[a]	The message link server of claim 1, wherein the transport services stack is further to receive, over each of the respective secure message links, upload messages forwarded by the respective device link agents from at least a subset of the device software components,

Reference	Limitation
5[b]	each of the upload messages identifying a corresponding one of the network elements to which the device respective software component has requested delivery,
5[c]	the network server system using the interface to a network to deliver content from the upload messages to the respective identified network elements.
Claim 6	
6	The message link server of claim 1, wherein at least one of the one or more messages for delivery by the transport services stack comprises multiple identifier/data pairs.
Claim 7	
7	The message link server of claim 1, the device messaging agent on at least one of the wireless end-user devices further to initiate the respective secure Internet data message link to the transport services stack.
Claim 8	
8	The message link server of claim 1, further comprising a secure server to provide secure authorization signatures to the given one of the wireless end-user devices, the secure authorization signatures indicating the authorized software components that are allowed to receive data from secure message link messages via the message link server.
Claim 9	
9	The message link server of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.

Reference	Limitation
Claim 10	
10	The message link server of claim 9, wherein the period of the timer is fractionally shorter than a maximum data message interval beyond which the secure message link is taken down.
Claim 11	
11	The message link server of claim 1, wherein one of the message delivery triggers is the receipt of a transmission on the respective secure 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 12	
12	The message link server of claim 11, wherein the transmission is a heartbeat message generated by the given device link agent, or a request received from the given device link agent.
Claim 13	
13	The message link server of claim 1, wherein one of the message delivery triggers is the receipt of a particular network element message from one of the network elements.
Claim 14	
14	The message link server of claim 1, wherein one of the message delivery triggers is based on an amount of wireless network data usage consumed by the given one of the wireless end-user devices.
Claim 15	
15[pre]	A method of operating a message link server, comprising:
15[a]	maintaining a respective secure message link through an Internet network between the message link server and a respective device link agent on each of a plurality of wireless end-user devices,

Reference	Limitation
15[b]	each of the wireless end-user devices comprising multiple software components authorized to receive and process data from secure message link messages received via a device link agent on that device;
15[c1]	receiving network element messages from a plurality of network elements,
15[c2]	the received network element messages comprising respective message content and requests for delivery of the respective message content to respective wireless end-user devices, the respective message content including data for, and an identification of, a respective one of the authorized software components;
15[d1]	buffering content from the received network element messages for which delivery is requested to a given one of the wireless end-user devices;
15[d2]	determining when one of a plurality of message delivery triggers for the 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 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
15[d3]	upon determining that one of the message delivery triggers has occurred, supplying one or more messages comprising the buffered content for delivery on the secure message link maintained between the message link server and a device link agent on the given one of the wireless end-user devices.

Amazon.com Services LLC and Amazon Web Services, Inc. ("Amazon" or "Petitioners") petition for *inter partes* review ("IPR") of claims 1-15 ("Challenged Claims") of U.S. Patent 9,615,192 ("'192Pat").

I. IPR REQUIREMENTS

A. Grounds for Standing

Petitioners certify the '192Pat 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 '192Pat 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)
Rakic (Ex-1015)	Filed 5/12/2008	102(e)
Herzog (Ex-1008)	Filed 12/17/2007	102(e)
Gellens (Ex-1020)	Published 6/30/2005	102(b)
Adamczyk (Ex-1021)	Published 7/3/2008	102(a)/(e)

2. Grounds

Petitioners request cancellation of the Challenged Claims based on the following grounds.

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

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

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

II. ’192 PATENT

The ’192Pat’s “message link server” 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. ’192Pat, Abstract; Traynor, ¶¶49-90.

To transmit messages to end-user devices, a secure link is established between the “message link server” (red) and a “device link agent” (blue) on the end-user’s device. ’192Pat, Abstract, 89:11-90:4; Traynor, ¶70.

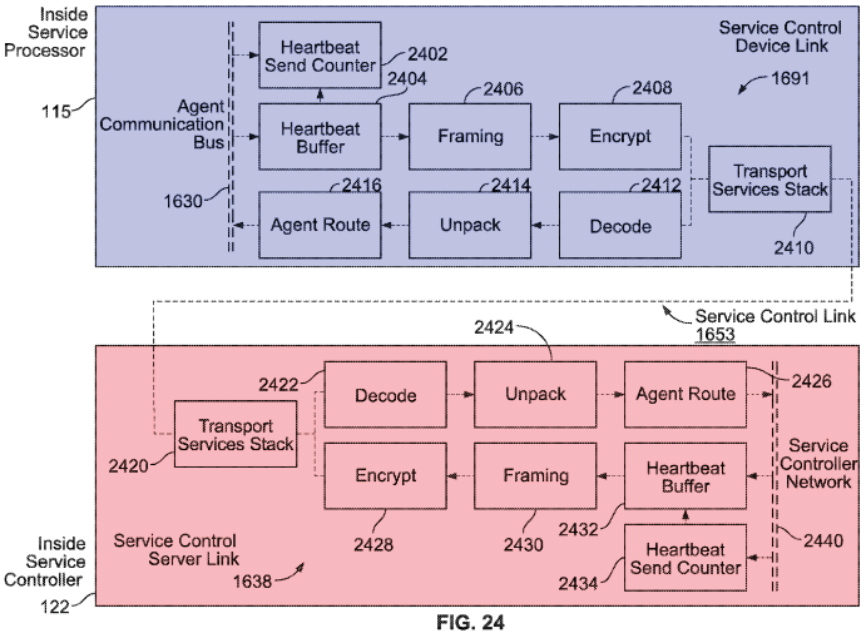


FIG. 24

’192Pat, Fig. 24.[4]

[4] Color annotations added unless otherwise noted.

III. NO PRIOR IPR OR DISTRICT COURT HAS RULED ON INVALIDITY OF THE '192PAT

In March 2023, Headwater Research LLC ("PO") sued Samsung in district court alleging infringement of the '192Pat. Ex-1029. PO withdrew its allegations regarding the '192Pat before trial, and the court dismissed all claims/causes of action regarding the '192Pat with prejudice on May 1, 2025. Ex-1022. The court did not rule on the invalidity of any Challenged Claim under §§102/103.

In November 2023, Samsung filed a petition for IPR of claims 1-9, 11-13, and 15 of the '192Pat. *Samsung Elecs. Co. v. Headwater Rsch. LLC*, IPR2024-00010, Paper 2 (Nov. 17, 2023). In May 2024, the Board instituted that IPR, and an oral hearing was held in March 2025. *Id.*, Papers 7, 24. Subsequently, Samsung and PO settled and moved to terminate the IPR. *Id.*, Paper 25. The Board terminated the IPR without issuing a final written decision. *Id.*, Paper 27.

IV. LEVEL OF ORDINARY SKILL IN THE ART

Persons of ordinary skill in the art ("POSITAs") relating to the '192Pat'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.*

V. CLAIM CONSTRUCTION

A District Court construed "software components" as "components of software" and claim 13's "wherein" clause as having its plain and ordinary meaning. Ex-1023, 16-19.

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 '192Pat specification/'192FH inform certain terms' ordinary meanings, Petitioners address this in the grounds below.

VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Ground 1A: TS-23.140 (Claims 1, 5-7, 9, 11-13, 15)

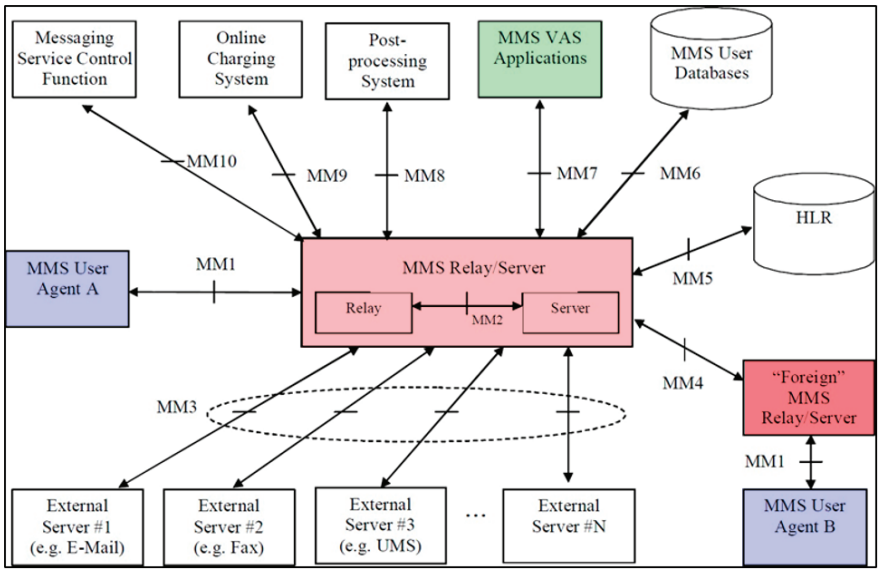
1. TS-23.140

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, ¶¶99-107.

---

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

[6] Citations refer to publication page number.



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 ("User Agents") on end-user devices (blue), out-of-network MMS 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. Claims 1, 15

a. 1[pre]/15[pre]

If limiting, TS-23.140 discloses/suggests 1[pre]/15[pre]. Traynor, ¶¶139-146.

TS-23.140's Relay/Server (*message-link server*[7]) facilitates message delivery over a network. TS-23.140, 14, 17, 54-55; Traynor, ¶140.

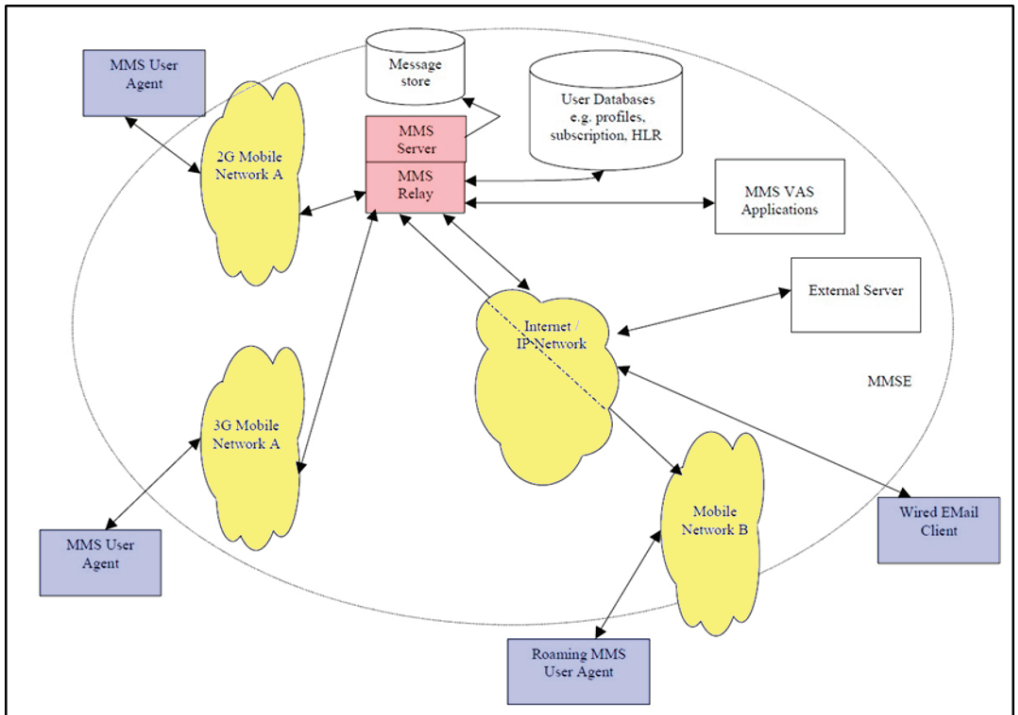


Figure 2: MMS Architectural Elements

TS-23.140, Fig. 2.

The Relay/Server and method for operating it are detailed in §§VI.A.2.b-VI.A.2.h (1[a]-[d4]/15[a]-[d3]).

b. 1[a]/15[a]

TS-23.140 discloses/suggests 1[a]/15[a]. Traynor, ¶¶147-60.

[7] Italics indicate claim language.

TS-23.140's Relay/Server establishes/maintains a link over an Internet network (yellow) with User Agents executing on each user equipment ("UE")/mobile device. TS-23.140, 17, 19-20; Traynor, ¶148; Ex-1010, Fig. 3.

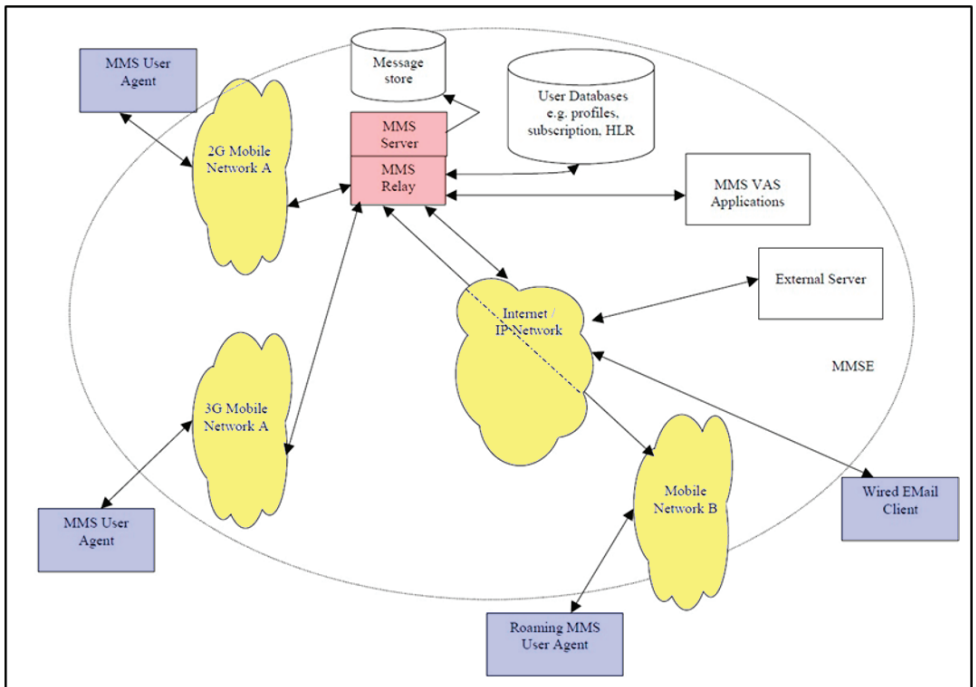
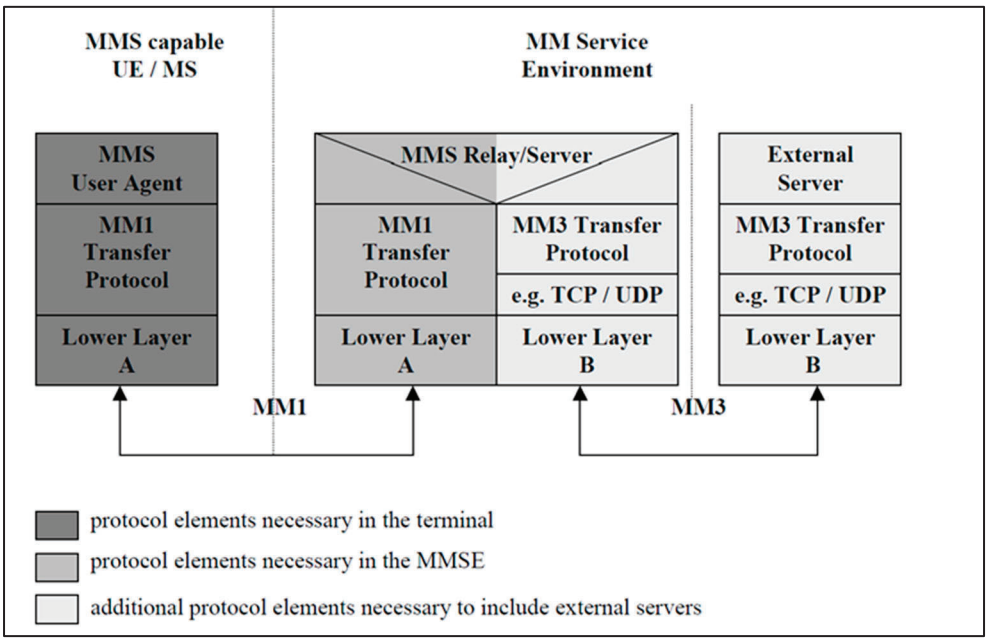


Figure 2: MMS Architectural Elements

TS-23.140, Fig. 2.

In TS-23.140, (1) Relay/Server and User Agent utilize a transport stack 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, ¶158 (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 that "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, ¶¶156-57.

[8] Emphases added unless otherwise noted.

POSITAs would have understood/found obvious TS-23.140 discloses a *transport-services stack* comprising MM1 Transfer Protocol with TCP/IP and TLS protocols. Traynor, ¶¶151-55. Indeed, this is consistent with the '192Pat's description of a *transport-services stack*. *See* '192Pat, 89:24-41, 90:34-50; Traynor, ¶¶159-60 (citing Munson).

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, ¶¶159-60 (citing Mostafa, 2-3, Figs. 1, 3; Munson, Fig. 1, [0007]-[0008]; Houghton, 23).

Accordingly, TS-23.140 discloses/suggests Relay/Server (*message link server*) comprises MM1 Transfer Protocol with TCP/IP and TLS (*transport services stack*) to communicate with UEs/mobile devices. Traynor, ¶¶147-60. Further, TS-23.140 discloses/suggests the *transport services stack* maintains a TLS-secured communication link over the Internet (*maintain[ing] a respective secure message link through an Internet network*) *between* Relay/Server (*message link server*) and a User Agent (*respective device link agent*) on each of the UEs/mobile devices (*on each of a plurality of wireless end-user devices*). *Id.*

c. 1[b]/15[b]

TS-23.140 discloses/suggests 1[b]/15[b]. Traynor, ¶¶161-72.

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 and process application-specific MMs via MMS on each UE/mobile device, which was common by 2009. TS-23.140, 54-55; Traynor, ¶170 (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, ¶¶163-69. Once registered, Relay/Server may deliver application-specific MMs using protocols including TLS to the registered application. TS-23.140, 30, 54-55; Traynor, ¶170; §VI.A.2.b (1[a]/15[a]).

POSITAs would have further understood/found obvious that application-specific MMs *received* by a particular application would be *processed* by that application. Traynor, ¶¶161, 166, 171; TS-23.140, 56.

Because application-specific MMs are transmitted to/from User Agents through Relay/Server (§VI.A.2.b (1[a]/15[a])), POSITAs would have understood MMs are received via User Agent on the particular UE/mobile device. Traynor, ¶¶169-70. Moreover, because MMs may be transmitted over TLS (a *secure-message*

link, §VI.A.2.b (1[a]/15[a])), they constitute *secure message link messages*. Traynor, ¶¶151-53, 155-59.

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/process data from* application-specific MMs that User Agent received from Relay/Server via a TLS-secured communication link (*secure message link messages received via a device link agent on that device*). Traynor, ¶¶161-72.

d. 1[c1]/15[c1]

TS-23.140 discloses/suggests 1[c1]/15[c1]. Traynor, ¶¶173-80.

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, ¶¶174-75; §§VI.A.2.b (1[a]/15[a]), VI.A.2.c (1[b]/15[b]) (Relay/Server receives messages from User Agents/VAS Applications).

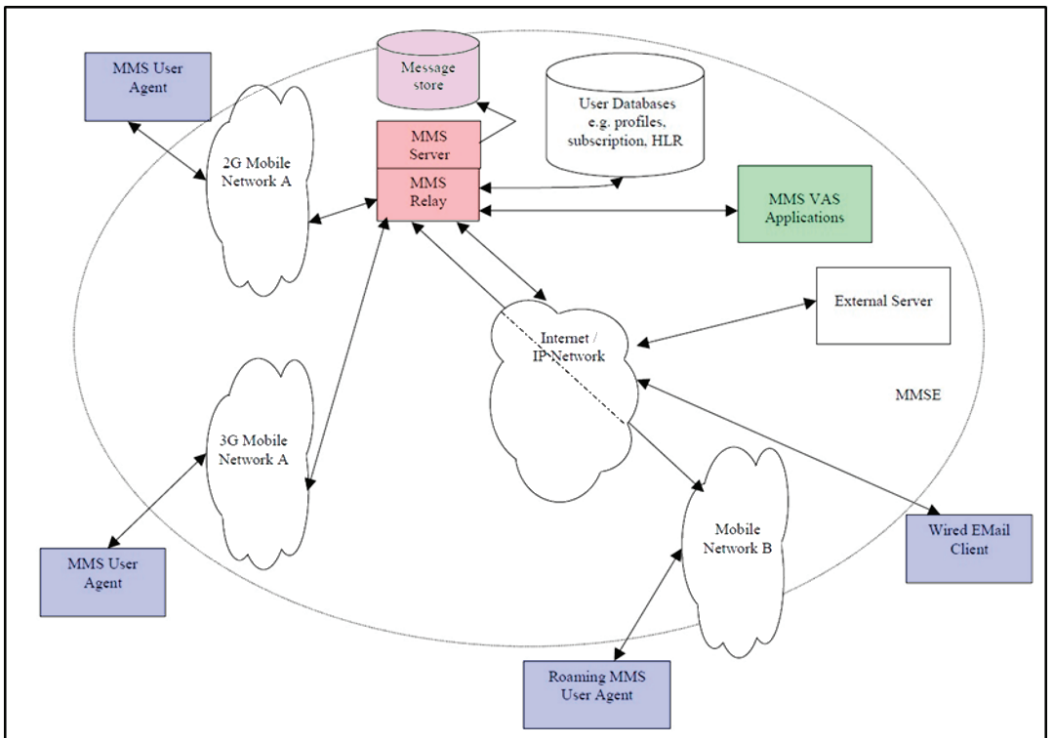
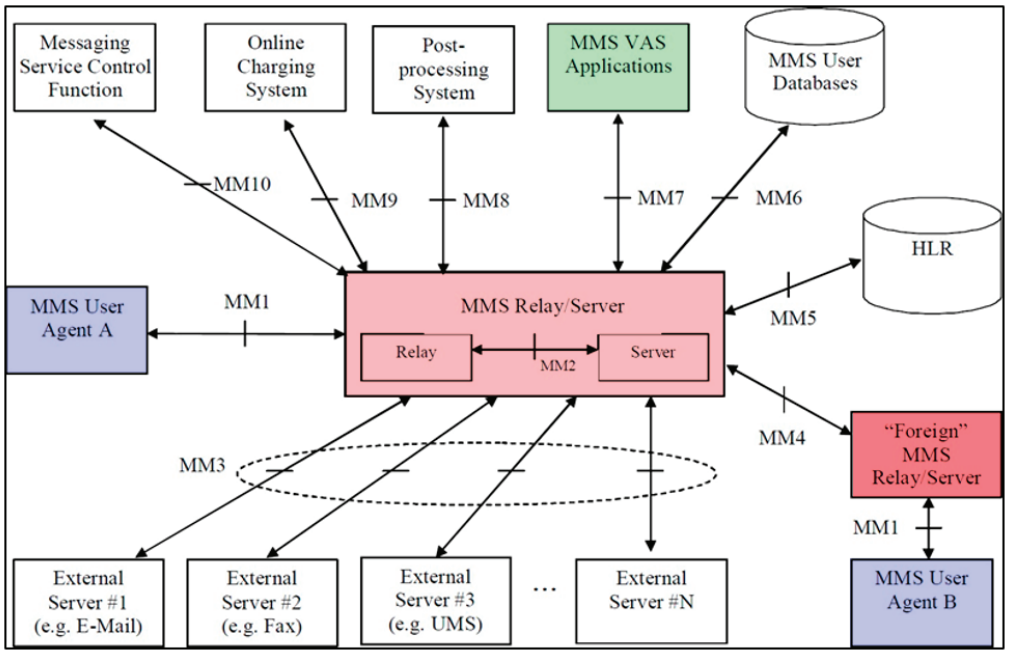


Figure 2: MMS Architectural Elements

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, ¶¶176-78.

Registered applications transmit application-specific MMs to other UEs or application servers (on other *network elements*) via Relay/Server. §VI.A.2.c (1[b]/15[b]); TS-23.140, 54-55; Traynor, ¶179.

Accordingly, TS-23.140 discloses/suggests the Relay/Server communicating using MM1/MM4/MM7 protocols over 2G/3G/IP/internet networks (*interface[s] to a network*) and receiving messages from, e.g., User Agents and VAS Applications (*receiv[ing] network-element messages from a plurality of network elements*). Traynor, ¶¶173-80.

e. 1[c2]/15[c2]

TS-23.140 discloses/suggests 1[c2]/15[c2]. Traynor, ¶¶181-88.

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, ¶¶182-83; §§VI.A.2.b (1[a]/15[a]), VI.A.2.c (1[b]/15[b]).

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, ¶¶183-86.

Recipient/destination applications teach authorized/registered applications (§VI.A.2.c, (1[b]/15[b])) because applications "need to register" using their "application identification value" to utilize MMS. TS-23.140, 54-56; Traynor, ¶186. 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 messages and passes "on the destination application identifier" and "application data" to, e.g., another User Agent. TS-23.140, 54-56; §VI.A.2.d (1[c1]/15[c1]); Traynor, ¶¶184-85.

Accordingly, TS-23.140 discloses/suggests Relay/Server receiving application-specific MMs (*received network element messages*) from a device (UE/mobile device, server) and its associated agent (User Agent, MMS VAS Application(s)) to another device and its associated agent/registered application. Traynor, ¶¶181-88. Further, the received application-specific MMs include *message content* comprising application data/control information (*data*) for the registered application and a destination "application identifier" (*identification*) of the registered

application (*message content including data for, and an identification of, a respective one of the authorized software components*). *Id.*, ¶¶184-86. Because an originating User Agent sends messages to the Relay/Server for delivery to a User Agent with the registered destination application, POSITAs would have understood or found obvious that the received application-specific MMs comprise *requests for delivery of the respective message content to respective wireless end-user devices* (including the destination application(s) resident on that device). *Id.*, ¶¶185, 187 (citing TS-23.140, 28-29, 54-55, Fig. 6).

f. 1[d1]-[d2]/15[d1]

TS-23.140 discloses/suggests 1[d1]-[d2]/15[d1]. Traynor, ¶¶189-96.

TS-23.140's Relay/Server receives network-element messages with requests to deliver the messages to other application(s) on other UEs/wireless devices. Traynor, ¶190; §§VI.A.2.d (1[c1]/15[c1]), VI.A.2.e (1[c2]/15[c2]).

In that process, Relay/Server "stores" messages and "handl[es]" transfer "between different messaging systems." TS-23.140, 17, 21; Traynor, ¶191.

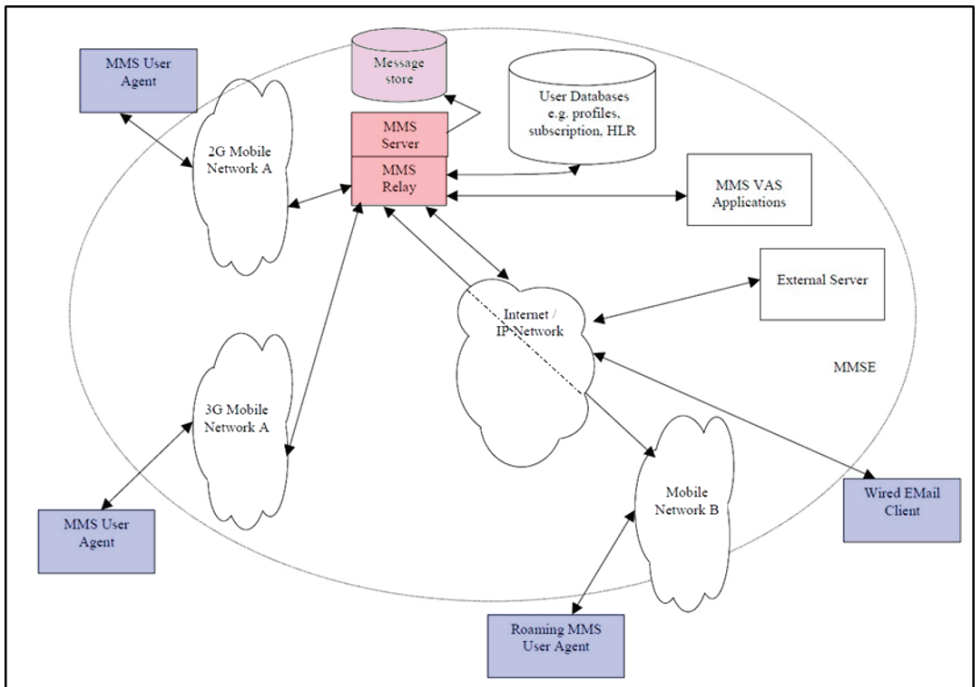


Figure 2: MMS Architectural Elements

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, ¶¶192-94.

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, ¶¶189-96;

§§VI.A.2.g-VI.A.2.h (1[d3]-[d4]/15[d2]-[d3]). 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/mobile device (*buffer[ing] content from the received network messages for which delivery is requested to a given one of the wireless end-user devices*). Traynor, ¶¶193-96.

g. 1[d3]/15[d2]

TS-23.140 discloses/suggests 1[d3]/15[d2]. Traynor, ¶¶197-209.

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, ¶199. Several conditions trigger User Agent to issue the retrieval request and secure delivery. Traynor, ¶¶199. 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) initiates a delivery attempt. *Id.*, 29.

If a "trigger" further 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 the 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 messages dictating whether they 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*]/[*determining*] *when one of a plurality of message delivery triggers for a given one of the wireless end-user devices has occurred*). Traynor, ¶201.

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, ¶202. 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, ¶¶200-02. And if a trigger is a condition after which the next step is attempted delivery (conditions (1)-(3)), the receipt of an MM—with/without a specified delivery-time—is not a trigger. Traynor, ¶¶200-02. Under this interpretation, none of the received MMs are triggers—thus, *at least some of the received network element messages* are not triggers. Traynor, ¶202.

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, ¶203. 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). Traynor, ¶200. 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 ¶¶200-03. Such manual retrieval is a trigger that is an *occurrence of an asynchronous event with time-critical messaging needs*—indeed, the '192Pat discloses a manual "user request" is a trigger. '192Pat, 38:50-63; Traynor, ¶¶204-08.

h. 1[d4]/15[d3]

TS-23.140 discloses/suggests 1[d4]/15[d3]. Traynor, ¶¶210-13.

TS-23.140's Relay/Server delivers stored messages in response to a trigger. Traynor, ¶211; TS-23.140, 28-31; §§VI.A.2.f-VI.A.2.g (1[d1]-[d3]/15[d1]-[d2]). POSITAs would have found obvious that Relay/Server includes logic for attempting delivery upon a trigger event. §VI.A.2.g (1[d3]/15[d2]); Traynor, ¶¶210-12.

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. §VI.A.2.b (1[a]/15[a]); TS-23.140, 24, Fig. 4; Ex-1010, 22; Traynor, ¶212.

Accordingly, TS-23.140 discloses/suggests once Relay/Server detects a trigger (*upon determining that one of the message delivery triggers has occurred*), Relay/Server includes logic to supply MMs stored in its Message Store/MMBox to MM1 Transfer Protocol with TCP/IP and TLS ([*the logic further to supply*/*supplying*] *one or messages comprising the buffered content* [*to the transport services stack*]). Traynor, ¶¶210-11. 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/message link server*] *and a device link agent on the given one of the wireless end-user devices*). *Id.*, ¶212.

3. Claim 5

a. 5[a]

TS-23.140 discloses/suggests 5[a]. Traynor, ¶¶214-21.

In TS-23.140, an application on a UE/mobile device (*device-software component*) forwards an application-specific MM (*upload message*) to UE's User Agent (*respective device-link agent*) for sending to Relay/Server. TS-23.140, 14, 54-55; Traynor, ¶¶215-16. TS-23.140's User Agent can submit/forward to Relay/Server MMs (*upload messages*) with requests to store them. TS-23.140, 40.

POSITAs would have understood/found obvious that Relay/Server "transports" "application data" between a User Agent and another User Agent/VAS Application. TS-23.140, 19-21, 35, 55; Traynor, ¶¶217-19. Relay/Server receives/transports application data to/from the UE/mobile device via the MM1 Transfer Protocol with TCP/IP and TLS (*transport services stack*). §VI.A.2.b (1[a]/15[a]); TS-23.140, 24, Fig. 4; Traynor, ¶219.

POSITAs would have also understood multiple UEs/mobile devices transmit data to multiple network elements through the Relay/Server, and thus the Relay/Server receives upload messages (including application-specific MMs) from multiple User Agents over dedicated TLS-based links (*secure-message links*) using MM1 Transfer Protocols (part of the *transport-services stack*). §§VI.A.2.b-VI.A.2.c (1[a]-[b]/15[a]-[b]); Traynor, ¶¶159, 221 (citing Mostafa, 2-3, Figs. 1, 3; Houghton, 23; Munson, Fig. 1, [0007]-[0008]).

Accordingly, TS-23.140 discloses/suggests Relay/Server receives messages via the MM1 Transfer Protocol with TCP/IP and TLS-secured links (*transport*

services stack is further to receive, over each of the respective secure message links), where those messages are submitted by/forwarded from a User Agent corresponding to at least one of the applications registered therewith (*upload messages forwarded by the respective device link agents from at least a subset of the device software components*). Traynor, ¶¶215-21.

b. 5[b]

TS-23.140 discloses/suggests 5[b]. Traynor, ¶¶222-28.

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 the "application identifier of the destination application." TS-23.140, 54-55; §VI.A.3.a (5[a]). Moreover, originating User Agent "indicate[s]" the message recipient's address. TS-23.140, 26, 90, 190; Traynor, ¶¶223-24. Because the message includes an identifier of a destination application on a particular UE/mobile device and/or indicates the "recipient address," POSITAs would have understood/found obvious that 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, ¶¶225-226.

POSITAs would have recognized routing messages to particular devices is facilitated by including device identification. Traynor, ¶¶225-26. 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, ¶¶225-28.

Accordingly, TS-23.140 discloses/suggests Relay/Server uses the MM1 Transfer Protocol with TCP/IP and TLS to receive upload messages from a User Agent and those 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, ¶¶222-28.

c. 5[c]

TS-23.140 discloses/suggests 5[c]. Traynor, ¶¶229-30.

TS-23.140's Relay/Server (*network-server system*)[9] uses its MM1, MM2, MM7, and MM4 interfaces (*interface*) to 2G, 3G, and IP/internet networks (*network*)

[9] Claim 5's *the network server system* lacks antecedent basis. The analysis considers the scope as including claim 1's *message-link server* and its network *interface*. Traynor, ¶229 n.5. Petitioners reserve the right to argue this term is indefinite in other proceedings.

to deliver the upload messages (*content*) to the destinations (*respective identified network elements*). §§VI.A.2.b-VI.A.2.d (1[a]-[c1]/15[a]-[c1])); Traynor, ¶¶229-30.

4. Claim 6

TS-23.140 discloses/suggests claim 6. Traynor, ¶¶231-36.

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*). §§VI.A.2.b (1[a]/15[a]), VI.A.2.d-VI.A.2.e (1[c1]-[c2]/15[c1]-[c2]), VI.A.2.g-VI.A.2.h (1[d3]-[d4]/15[d2]-[d3]); TS-23.140, 54-56; Traynor, ¶¶232-33.

POSITAs would have understood multiple registered applications on a UE/mobile device receive messages including application data via Relay/Server. §§VI.A.2.d-VI.A.2.e (1[c1]-[c2]/15[c1]-[c2]); TS-23.140, 54-56; Traynor, ¶233 (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, ¶234. For instance, message delivery is triggered when User Agent becomes available/reachable (TS-23.140, 28-30), meaning multiple messages may be queued. Traynor, ¶234. 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, ¶234.

Accordingly, TS-23.140 discloses/suggests Relay/Server (*message-link server*) using MM1 Transfer Protocol with TCP/IP and TLS (*transport-services stack*) to send a message comprising identifiers/data from one or more applications (*multiple identifier/data pairs*). Traynor, ¶¶231-36.

5. Claim 7

TS-23.140 discloses/suggests claim 7. Traynor, ¶¶237-41.

TS-23.140's Relay/Server uses MM1 Transfer Protocol with TCP/IP and TLS (*transport-services stack*) to transmit/receive messages over a TLS-secured link (*secure Internet data-message link*) between User Agent (*device-messaging agent on wireless end-user devices*) and Relay/Server. §VI.A.2.b (1[a]/15[a]); Traynor, ¶¶237-39. POSITAs would have found it obvious for the UE/mobile devices to initiate the TLS-secured communication link. Traynor, ¶237. Before January 28, 2009, it was well known for a client to initiate TLS-based communications, and this would have amounted to choosing from two predictable solutions (the Relay/Server or the User Agent initiating the link) with a reasonable expectation of success. Traynor, ¶¶238-41 (citing Ex-1012, 29:31-30:24).

6. Claim 9

TS-23.140 discloses/suggests claim 9. Traynor, ¶¶242-47.

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, ¶¶242-43. 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, ¶¶242-44.

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, ¶¶245-46. 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 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, ¶246.

7. Claims 11-12

TS-23.140 discloses/suggests claims 11-12. Traynor, ¶¶248-49.

TS-23.140 triggers delivery by having User Agent (*device-link agent*) manually request delivery from Relay/Server over a TLS-secured link (*secure message link*) in response to a notification from Relay/Server (*receipt of a transmission on the respective secure 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). §§VI.A.2.b (1[a]/15[a]), VI.A.2.f (1[d1]-[d2]/15[d1]), VI.A.2.g (1[d3]/15[d2]); TS-23.140, 28-29; Traynor, ¶248.

8. Claim 13

TS-23.140 discloses/suggests claim 13. Traynor, ¶¶250-52.

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 network element message from one of the network elements*). §VI.A.2.g (1[d3]/15[d2]); Traynor, ¶¶250-51.

B. Ground 1B: TS-23.140-Shen (Claims 2-3, 12)

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, ¶¶108-14. To address this, Shen proposes encrypting the messages using keys/certificates (dark yellow). Shen, Fig. 1, [0021], [0030], [0054]-[0060].

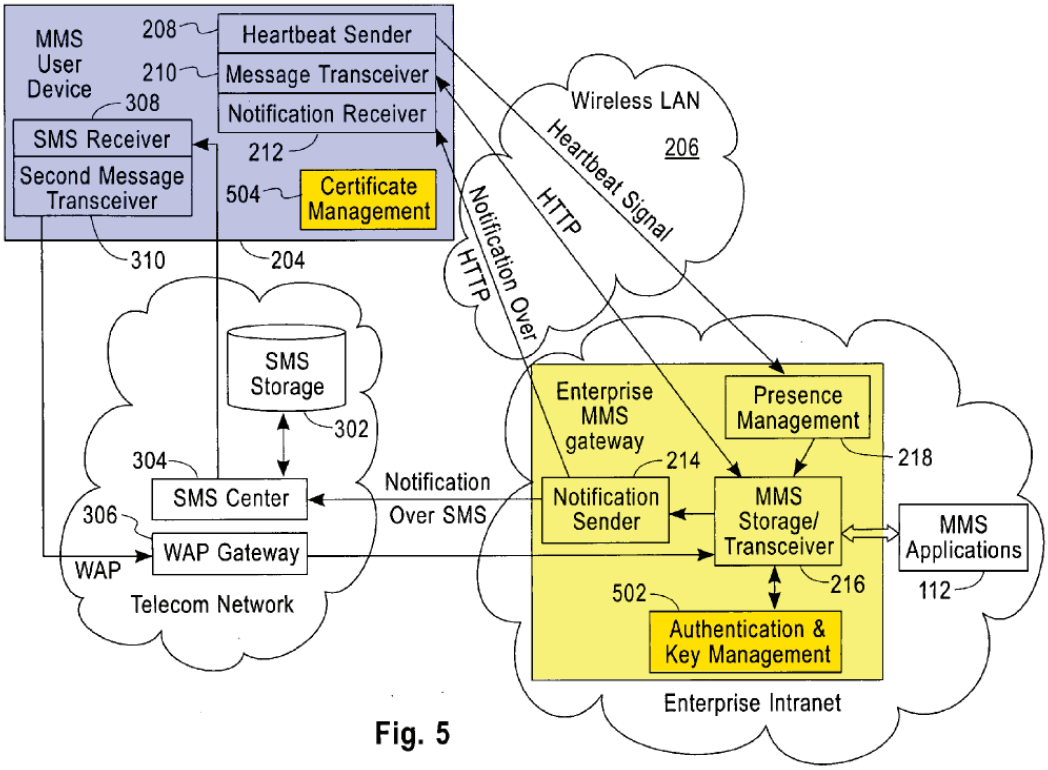


Fig. 5

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].

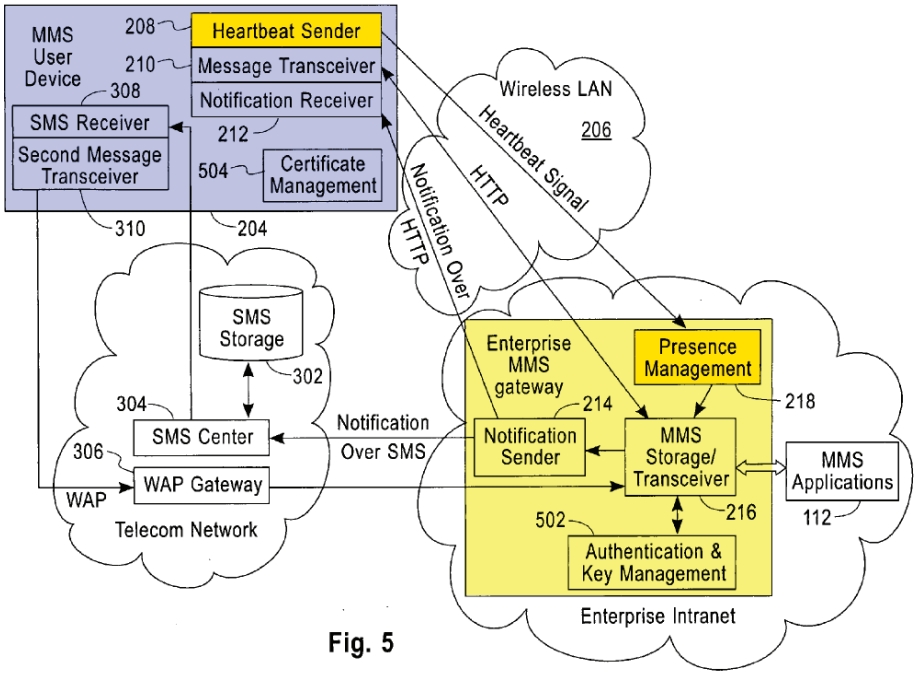


Fig. 5

Shen, Fig. 5.

2. Claim 2

TS-23.140-Shen teaches claim 2. Traynor, ¶¶253-64.

TS-23.140 delivers messages through MM1 Transfer Protocol with TCP/IP and TLS (*transport-services stack*) over a TLS link (*secure-message link*) between Relay/Server (*message-link server*) and UE/mobile device's User Agent (*device-link agent on a wireless end-user device*). §§VI.A.2.b (1[a]/15[a]), VI.A.2.h (1[d4]/15[d3]); Traynor, ¶¶253-54. Although TS-23.140 teaches encrypting the connection, it does not explicitly disclose encrypting messages. Traynor, ¶254.

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 (*encryption function to encrypt one or more messages*) at Relay/Server (*message-link server*) for "encrypting MMS messages." §VI.B.1 (Shen); Shen, [0004], [0029]-[0034], [0059], Fig. 5; Traynor, ¶¶255-58. Shen's messages can be "encode[d] and decode[d] to protect the privacy of these messages." Shen, [0054]; Traynor, ¶257.

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" at Relay/Server that generates and "distribute[s] symmetric keys to users" and uses the symmetric keys to encrypt data transmitted to "user device[s]" (Shen, [0054]-[0060])—to improve security by preventing unauthorized access to Relay/Server's stored messages. Traynor, ¶258. 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, ¶¶258-60.

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, ¶261. 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, ¶¶262-63.

Accordingly, TS-23.140-Shen teaches *an encrypt function to encrypt one or more messages supplied to the* MM1 Transfer Protocol with TCP/IP and TLS (*transport-services stack*) *for delivery on the* TLS link (*secure-message link*) *maintained between the* Relay/Server (*message-link server*) and UE/mobile device's User Agent (*the device-link agent on the given one of the wireless end-user device*). Traynor, ¶¶253-64.

3. Claim 3

TS-23.140-Shen teaches claim 3. Traynor, ¶¶265-68.

TS-23.140-Shen teaches the *encrypted one or more messages* (§VI.B.2 (cl. 2)) *are transported to the* User Agent (*device-link agent*) *on the given one of the wireless end-user devices using* MM1 Transfer Protocol with TCP/IP and TLS (*encryption on the transport services stack*). §VI.A.2.b (1[a]/15[a]); Traynor, ¶267.

4. Claim 12

TS-23.140-Shen teaches claim 12. Traynor, ¶¶269-74.

Shen's user device generates a *heartbeat message* to indicate device reachability. Shen, [0033]-[0034]; Traynor, ¶271.

POSITAs would have been motivated to incorporate Shen's "heartbeat" as an additional trigger in TS-23.140. Traynor, ¶¶272-73. Specifically, one TS-23.140 trigger occurs when a 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, ¶¶269-71. 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, ¶¶272-73.

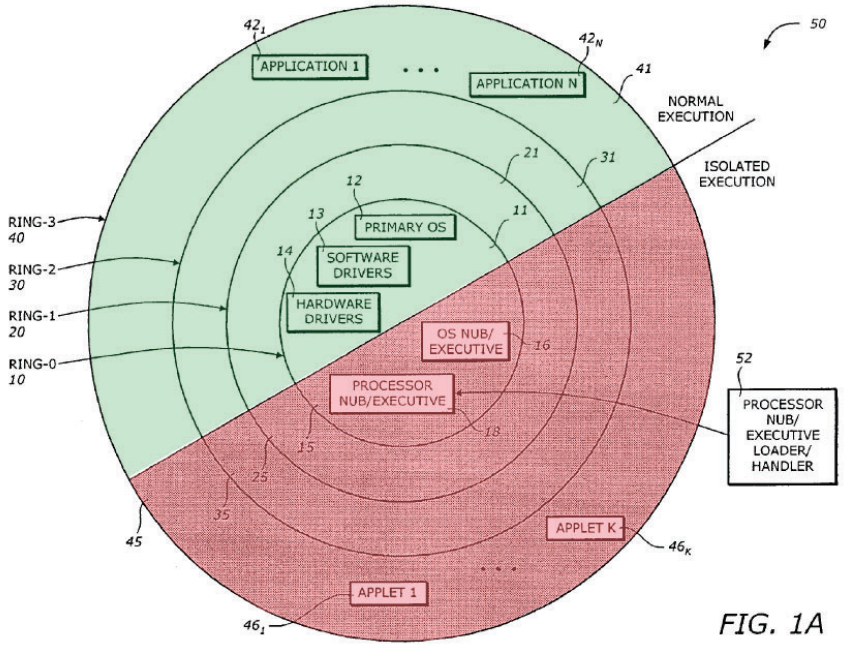
POSITAs would have had a reasonable expectation of success in doing so because it merely would have involved configuring a messaging signal from User Agent to Relay/Server indicating agent's availability/reachability, and Shen already discloses the contents of that message. Shen, [0034]; Traynor, ¶273.

C. Ground 1C: TS-23.140-Ellison (Claim 4)

TS-23.140-Ellison teaches claim 4. Traynor, ¶¶275-94.

TS-23.140 discloses both User Agent (*device-link agent*) and applications executing on UE/mobile device (*one of the devices*) (TS-23.140, 17-18, 23), but does not expressly disclose executing User Agent in a secure environment. Traynor, ¶276.

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 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, ¶¶277-79.



Ellison, Fig. 1A.

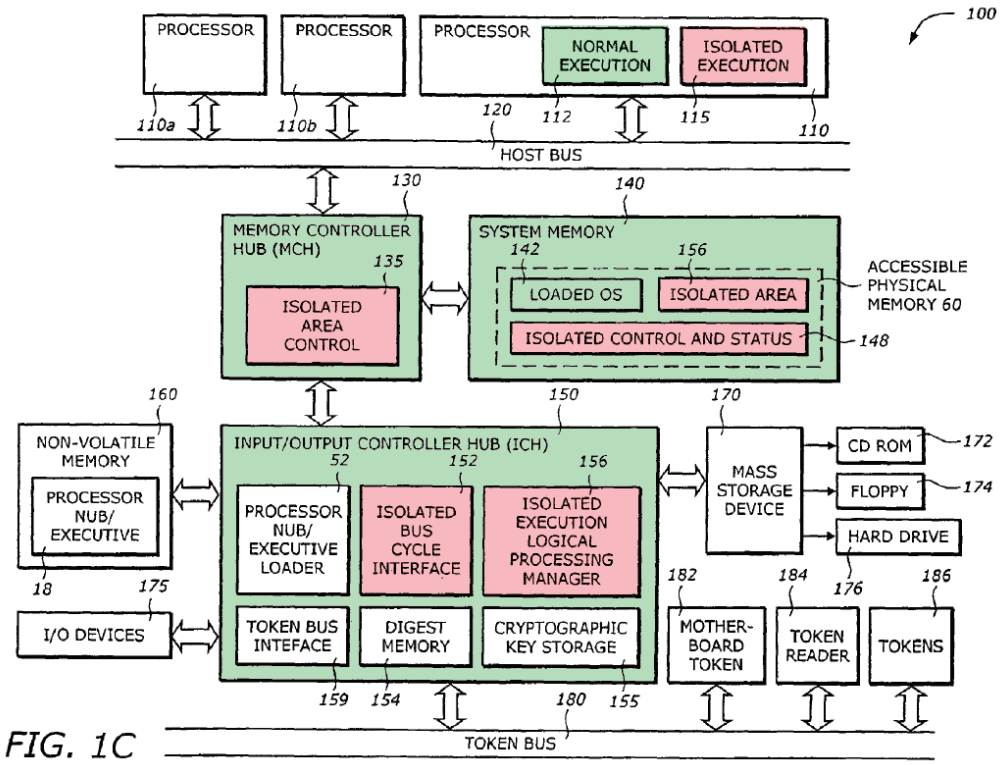


FIG. 1C

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, ¶¶280-81. 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, ¶288. 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, ¶¶287-88. This would have involved combining prior-art elements according to known methods to yield predictable results. Traynor, ¶¶288-291.

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, ¶¶280-87 (citing Ex-1033).

Accordingly, TS-23.140-Ellison teaches User Agent (*device-link agent*) 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 executes outside of the secure execution environment on that device*). Traynor, ¶¶275-94.

D. Ground 1D: TS-23.140-Rakic (Claim 8)

TS-23.140-Rakic teaches claim 8. Traynor, ¶¶295-314.

TS-23.140's User Agents operating on UEs/mobile devices receive messages over a TLS link via the MMS Relay/Server. TS-23.140, 23, 54-56. But TS-23.140 does not explicitly disclose a mechanism to verify the sender's identity or that the destination application is the intended recipient, thus leaving the User Agent exposed to potential attacks. TS-23.140, 54-56; Traynor, ¶¶296-302.

Rakic recognized the risk of unverified transmissions and proposed the use of electronic signatures to verify the parties to the transmission. Rakic, [0047]-[0049], [0066]-[0067]; Traynor, ¶¶303-05. Specifically, in Rakic, push-message server (504) generates an electronic signature, which is appended to the message and/or other information sent to the client device to verify message authenticity and source. Rakic, [0041], [0047]-[0049], [0066]-[0067], [0109]-[0112], Fig. 8. Client device (102) then uses secure-push-message client (404) to validate the message and sender using the electronic signature. *Id.*, [0109]-[0112], [0041]. Message validity is confirmed if the received and generated signature portions match. *Id.*, [0109]-[0112]; Traynor, ¶¶303-05.

POSITAs would have been motivated to implement a secure-push-message server and signature generation like Rakic with the Relay/Server to improve security for the client device and the underlying applications by enabling the client device to (1) "authenticate the sender" and (2) "verify that an intended recipient of the secure push message is the client device, and that the client device includes a correct

component." Rakic, [0041]; Traynor, ¶¶309-14. This would have amounted to using (1) a known technique (electronic signatures) to improve similar devices (computer systems) in the same way (added security through verification); and (2) combining prior art elements (electronic signatures/computer systems) per known methods (secure servers, databases, and electronic signatures) to yield predictable results (enhanced verification-based security). Traynor, ¶¶309-10.

POSITAs would have had a reasonable expectation of success in making this modification because TS-23.140 and Rakic contemplate similar messaging architectures/systems, and POSITAs would have found it straightforward to implement Rakic's teachings within a separate server or the Relay/Server. *Id.* Additionally, the resulting system's elements would perform functions performed prior to combination—the separate server or the Relay/Server would communicate messages to User Agents and other network elements, and Rakic's teachings (in combination) would enable providing secure signatures (with messages) to the receiving device(s). Traynor, ¶310.

Accordingly, TS-23.140-Rakic teaches a separate server or the Relay/Server that provides secure electronic signatures to the User Agent on a UE/mobile device (*a secure server to provide secure authorization signatures to the given one of the wireless end-user devices*) to indicate that a software component on the device is allowed to receive messages over the TLS-secured link via the Relay/Server (*the*

secure authorization signatures indicating the authorized software components that are allowed to receive data from secure message link messages via the message link server). Traynor, ¶¶295-314.

E. Ground 1E: TS-23.140-Adamczyk (Claim 9)

Ground 1A explains TS-23.140 discloses/suggests claim 9, which also would have been obvious over TS-23.140-Adamczyk. Traynor, ¶¶315-20.

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 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, ¶316.

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 ¶¶200, 317-18. 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.*, ¶319.

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. Traynor, ¶320. 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. Traynor, ¶320.

F. Ground 1F: TS-23.140-Herzog (With/ Without Adamczyk) (Claim 10)

TS-23.140-Herzog or TS-23.140-Adamczyk-Herzog teaches claim 10. Traynor, ¶¶321-26.

TS-23.140 and TS-23.140-Adamczyk teaches claim 9's periodic timer through periodic polling (TS-23.140) and a periodic message delivery schedule (Adamcyzk). §§VI.A.6 (cl.9), VI.E (cl.9); Traynor, ¶¶315-22. But these references do not explicitly disclose mechanisms to keep a connection active/inactive.

Herzog teaches mechanisms controlling connection duration between a 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 ¶323.

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, ¶324. 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 (IP, TCP, SSL, 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, ¶325.

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, ¶326. 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.*

G. Ground 1G: TS-23.140-Gellens (Claim 14)

TS-23.140-Gellen teaches claim 14. Traynor, ¶¶327-32.

TS-23.140 uses *message-delivery triggers* to control when messages are delivered to a User Agent. TS-23.140, 23, 30-31, 54-55; §§VI.A.2.d (1[c1]/15[c1]), VI.A.2.g (1[d][3]); Traynor, ¶¶327-28. TS-23.140 does not explicitly disclose triggering message delivery based on an amount of data consumed by the UE/mobile device. Traynor, ¶328.

Gellens recognized a need to control data delivery over cellular networks due to, e.g., costs and bandwidth, and permitting applications to open connections for "low priority" messages can drive up costs leading to "excessive and wasteful traffic channel usage." Gellens, [0004]; *see id.*, Abstract, [0002]-[0006], [0027], [0029]. To remedy this problem, Gellens proposes using triggers (like TS-23.140) to control when messages are delivered to clients, one of which triggers delivery of low-priority application messages only after a higher-priority application initiates a link. *Id.*, [0035]-[0036]; Traynor, ¶¶329-30.

POSITAs would have been motivated to combine TS-23.140 and Gellens to prioritize certain application messages, and only trigger transmission of low-priority application messages after a high-priority application opens a connection and is

consuming data to transmit its message (*one of the message delivery triggers is based on an amount of wireless network data usage consumed by the given one of the wireless end-user devices*). Traynor, ¶¶330-31. POSITAs would have recognized this would limit connection/transmission costs because "the cost of bringing the traffic channel up and down is spread over more traffic." Gellens, [0004], [0035]-[0036]; Traynor, ¶331. This would have amounted to applying a known technique (application prioritization) to a prior-art device (an MMS network) to solve a known problem (networking costs) and achieve a predictable result (triggering low-priority message delivery only after a high-priority application establishes a connection). *Id.*

POSITAs would have had a reasonable expectation of success in making this combination because it merely involves configuring the applications TS-23.140 already envisions being incorporated into the MMS network to be assigned priorities, and control what priorities are required to establish a connection between the Relay/Server and User Agent. Traynor, ¶332.

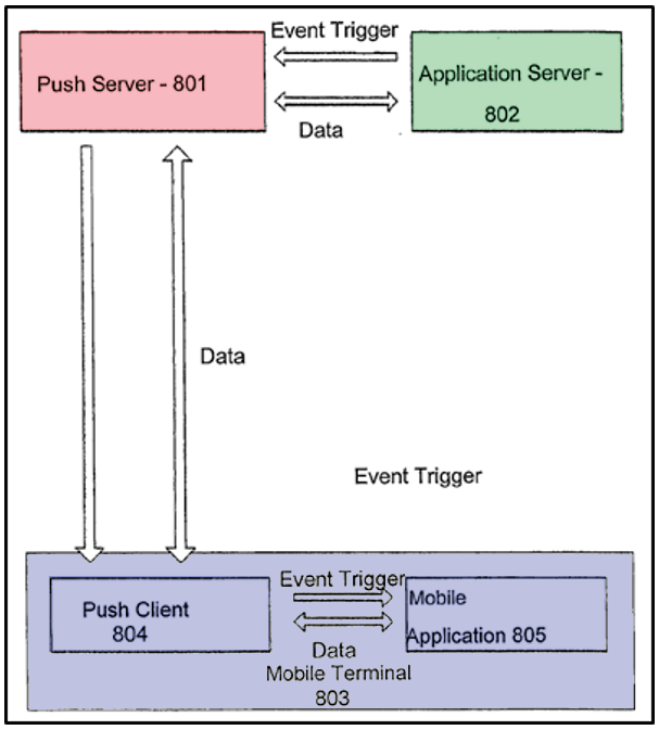
H. Ground 2A: Houghton-Munson (Claims 1, 5-7, 9-13, 15)

1. Houghton

Houghton's server "push[es] messages to a" client on a "mobile terminal" in a wireless network. Houghton, Abstract, 11[10]; Traynor, ¶122. Push messages "may

[10] Citations refer to publication page number.

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, ¶¶123-28



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:

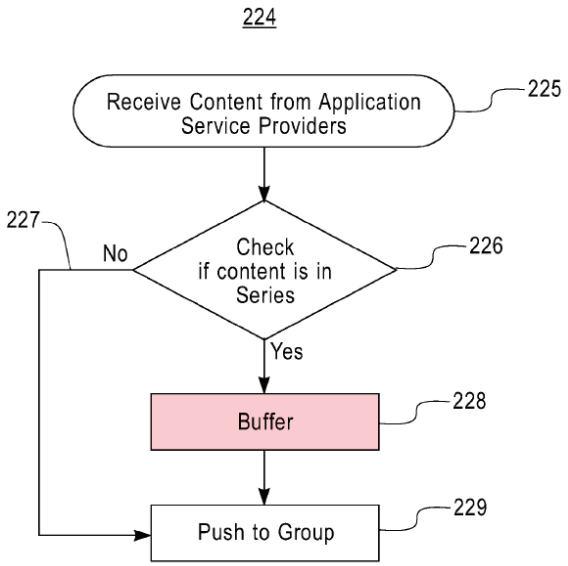


FIG. 4

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, ¶¶333-41.

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, ¶334. 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, ¶335; Houghton, 3, 7; Ex-1006, [0034], cl.14, Fig. 1); and (2) provide users greater options to control delivery times/conditions (Traynor, ¶336; 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, ¶337.

POSITAs would have had a reasonable expectation of success in doing so because Munson provides implementation details (*see, e.g.*, Munson, [0036]-[0038], [0044], Fig. 4) for "store and forward messaging systems" that Houghton contemplates (Houghton, 3, 7). Traynor, ¶338. 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, ¶¶338-40.

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.*, ¶341. Houghton discloses conditioning message delivery upon "any trigger event, local or remote." Houghton, 21; Traynor, ¶341.

Munson provides additional examples of message delivery triggers (Munson, [0040], [0044]), which would have required only routine programming within POSITAs' level of skill. Traynor, ¶341.

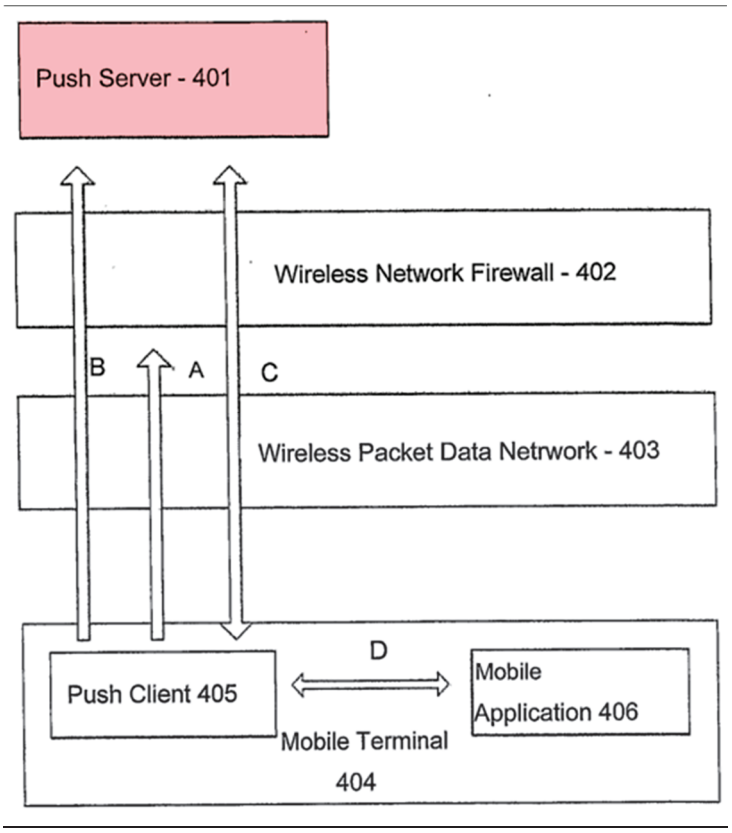
4. Claims 1, 15

a. 1[pre]/15[pre]

If limiting, Houghton discloses/suggests 1[pre]/15[pre]. Traynor, ¶¶342-345.

In Houghton, push server (401/801)[11] (*message link server*) sends push messages via communication links over a network to push client (405/804) on mobile terminals (404/803) for delivery to mobile applications (406/805). Houghton, Abstract, 16-17, Figs. 4, 8; Traynor, ¶¶342-43.

[11] POSITAs would have understood/found obvious to implement features regarding Houghton's Figure 4/6-8 embodiments together. Traynor, ¶342 n.6.



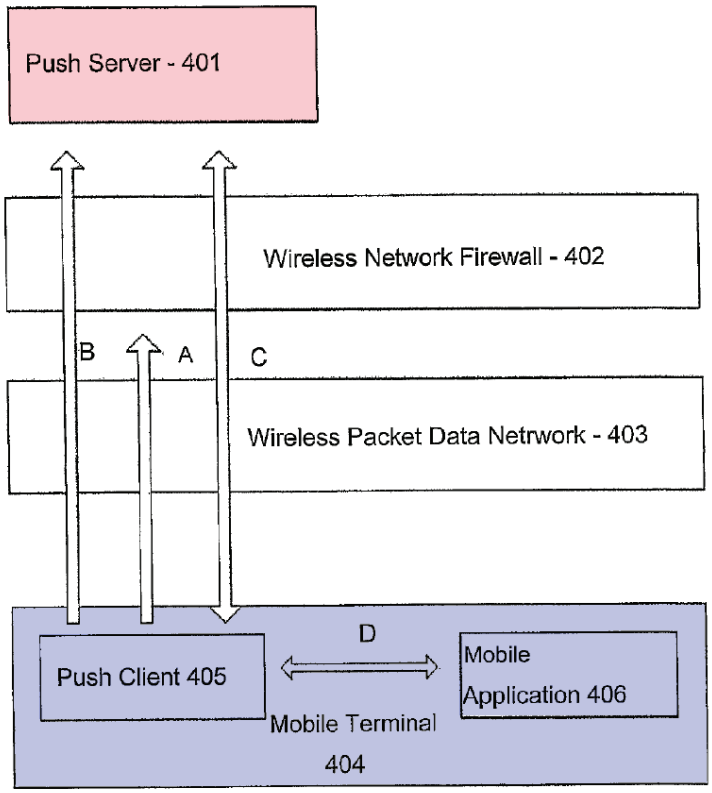
Houghton, Fig. 4.

Additional details regarding Houghton's push server, its combination with Munson, and the method for operating this server are described below. §§VI.H.4.b-VI.H.4.h (([1a-d4]/[15a-d3]); Traynor, ¶344.

b. 1[a]/15[a]

Houghton discloses/suggests 1[a]/15[a]. Traynor, ¶¶346-61.

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, ¶¶348-50.



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; '192Pat, 43:18-50; Traynor, ¶351.

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, ¶¶350-53. Thus, Houghton describes server-client communications using secure-transport protocols (*secure-message link*) like those in the '192Pat. Traynor, ¶354; '192Pat, 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 (*transport-services stack*) for establishing a link/transmitting messages over an Internet network with mobile terminal (404). Houghton, 1, 17-20, Fig. 1; Traynor, ¶354. Indeed, using protocol stacks for securely communicating over a network was well-known and conventional. Houghton, 1; Traynor, ¶¶354-56 (citing Ozaki, [0012]-[0022], Figs. 25-26). POSITAs would have understood Houghton discloses a "stack" of transport protocols. Traynor, ¶¶354, 357.

The communication link established between the push client/mobile terminal and push server (401) is *a secure-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, claim 10; Traynor, ¶358; '192Pat, 16:66-17:22, 39:20-32, 70:31-44. POSITAs would have understood/found obvious Houghton's push server's *transport-services stack* uses

such protocols for securing the TCP/IP-based communication link between push client (405/804) and push server (401/801). Traynor, ¶359.

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, ¶360; Houghton, 18-19 (multiple "terminals").

Accordingly, Houghton discloses/suggests push server using well-known TCP/IP transport protocols with SSL (*transport services stack*) *to maintain* a secure SSL or IPSec-based TCP/IP Internet connection (*secure message link*) *between* the push server (*message link server*) and push client on mobile terminals (*a respective device link agent on each of a plurality of wireless end-user devices*). Traynor, ¶¶346-61.

c. 1[b]/15[b]

Houghton discloses/suggests 1[b]/15[b]. Traynor, ¶¶362-82.

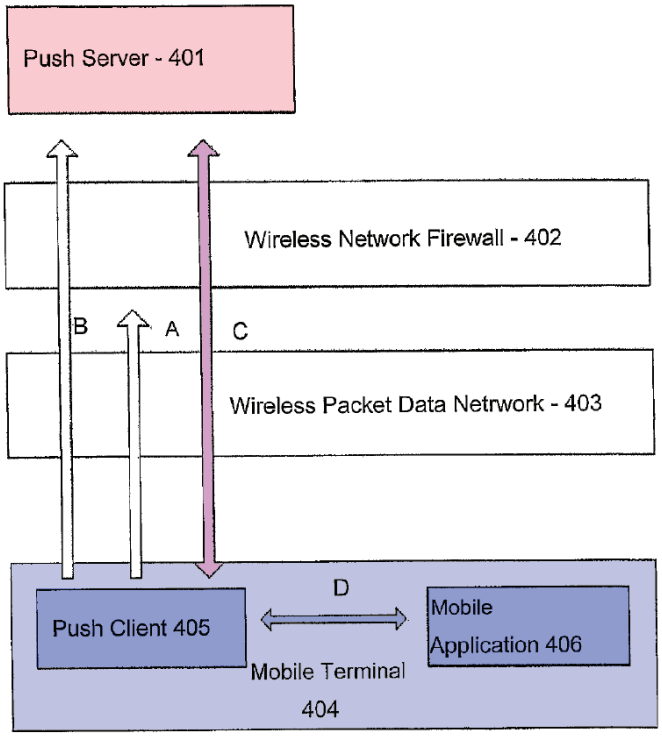
Mobile terminal (404) (*wireless end-user device*) includes a plurality of mobile applications (406) (*multiple software components*). Houghton, 21, 22, 25, 27, Figs. 4, 8; Traynor, ¶¶362-63.

Mobile applications (406) are configured to receive/process message data. Houghton, 21; Traynor, ¶366. Specifically, "application-specific data," e.g.,

"updates, commands or data messages" are "directed to a push application." Houghton, cl. 1, 11-12; Traynor, ¶¶367-69. 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, ¶370.

Push client (405) operating on mobile terminal (404) receives these messages over a secure TCP/IP link before being passing them to the relevant application (406). Traynor, ¶365. 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, ¶365; Houghton, 16-17 (push client (405) operates in software); '192Pat, 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, ¶371.



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, ¶372; 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, ¶373.

If Houghton does not expressly disclose authorizing applications to receive/process data from secure message link messages, the disclosed framework suggests it and POSITAs would have considered that obvious. Traynor, ¶¶374-80. Applications were well-known to register with a push server and/or push client before receiving messages via push frameworks. Traynor, ¶¶375-76 (citing Lee, [0023]; 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, ¶¶377-79. 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 devices users having to "seek out that data" and ensuring resource-efficient message delivery. Traynor, ¶¶378-79 (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, ¶¶377-80.

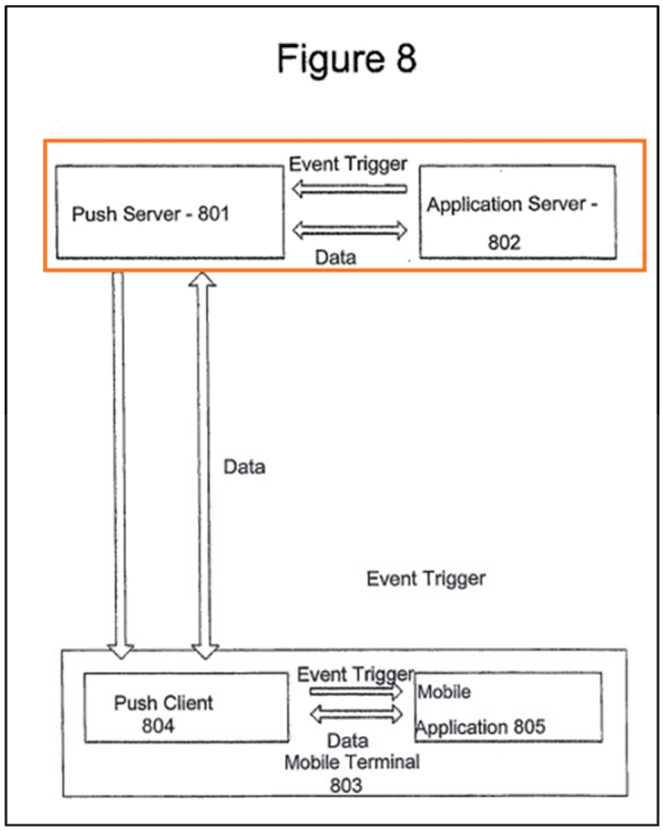
Accordingly, Houghton discloses/suggests mobile terminals (404) (*wireless end-user device*) each comprise multiple applications (406) (*multiple software*

components) *authorized* to receive and process application-specific data in messages received over a secure TCP/IP link (*secure message-link messages*) received by push client (405) (*device-link agent on that device*). Traynor, ¶¶362-82.

d. 1[c1]/15[c1]

Houghton discloses/suggests 1[c1]/15[c1]. Traynor, ¶¶383-93.

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, ¶¶384-88.



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, ¶¶384-88. 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, ¶¶390-91; Houghton, 21.

Accordingly, Houghton discloses/suggests an API (*interface to a network*) to receive application-specific messages (*network-element messages*) from application server(s) (e.g., 702, 802) (*plurality of network elements*). Traynor, ¶¶383-93.

e. 1[c2]/15[c2]

Houghton discloses/suggests 1[c2]/15[c2]. Traynor, ¶¶394-406.

Houghton's push server (401/801) receives push messages from an application server (e.g., 802) (*received network-element messages*) and sends them to one of a plurality of mobile applications (*authorized software components*) through push client (405/804) on mobile terminal (404/803). Houghton, 21-22; Traynor, ¶¶394-95. The message to the mobile applications is sent using "a data connection between application server 802 and mobile application 805," using the data connection (arrow C; Fig. 4) between push server 401/801 and push client 405/804. Houghton, 23; Traynor, ¶396.

Houghton discloses that messages sent from an application server to an application (through the push server) can take on several forms. Traynor, ¶397. The message may 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 (*identification of…the authorized software components*), and/or (3) "information to be passed to the…specified mobile application" (*data for…the authorized software components*). Houghton, 21; Traynor, ¶397. As one example, Houghton discloses a "trigger event" or "application command message" (*received network-element messages*) sent in response to a trigger (*request for delivery*), 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, ¶398. And because these messages are directed to "a mobile application 705 from a plurality of such applications" (*id.*), a POSITA would have understood that they also include *an identification of* the target mobile application. Traynor, ¶¶397-99.

To the extent Houghton is not found to explicitly disclose messages that include an *identification of* the mobile application, POSITAs would have found it obvious to include application identifiers in messages the application server sends and the push server receives. Traynor, ¶400. Before January 28, 2009, it was well-known to include an application identifier in messages directed to an application. *Id.* (citing Ex-1006, [0013], [0022] (push message includes application's "app_ID")). 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, ¶401. Moreover, including an application identifier when implementing Houghton's system 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, ¶402.

POSITAs would have also found obvious that the push message received by the push server includes a request for message content delivery (*requests for delivery*) to one or more mobile terminals. Traynor, ¶403. As discussed, Houghton explains "[p]ush server 701 is triggered by a trigger event…from an application server 702 to push an application command message." Houghton, 21-22. The trigger mechanism may be "an IP-triggered application launch and data transfer" e.g., a "mobile application launched by contact to a specified IP port." *Id.*, 22. "[P]ush server 901 is triggered by a trigger event of which the application server 902 is aware to send [a] push message to the mobile terminal 905," causing "pushing application launch commands, lazy application updates, and lazy application data updates through push clients 904 and 905 to mobile application 907." *Id.*, 24; Traynor, ¶404.

Accordingly, Houghton discloses/suggests the push server (*message-link server*) receives application-directed messages from an application server (*received network element messages*) that have *content and requests for delivery* of the contents to one of a plurality of applications on the mobile terminal (*wireless end-user devices*). Traynor, ¶¶394-405. Further, Houghton discloses/suggests the *content* in the message received by the push server includes data for, and an identifier of, the application authorized to receive the message (*the respective message content*

including data for, and an identification of, a respective one of the authorized software components). *Id.*

f. 1[d1]-[d2]/15[d1]

Houghton-Munson teaches 1[d1]-[d2]/15[d1]. Traynor, ¶¶407-17.

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. §§VI.H.4.d-VI.H.4.e (([1c1-c2]/[15c1-c2]), VI.H.4.g-VI.H.4.h ([1d3-d4]/[15d2-d3]); §VI.H.3 (motivation); Traynor, ¶¶408-11; Houghton, 3, 7.

Munson stores push messages in the 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, ¶413.

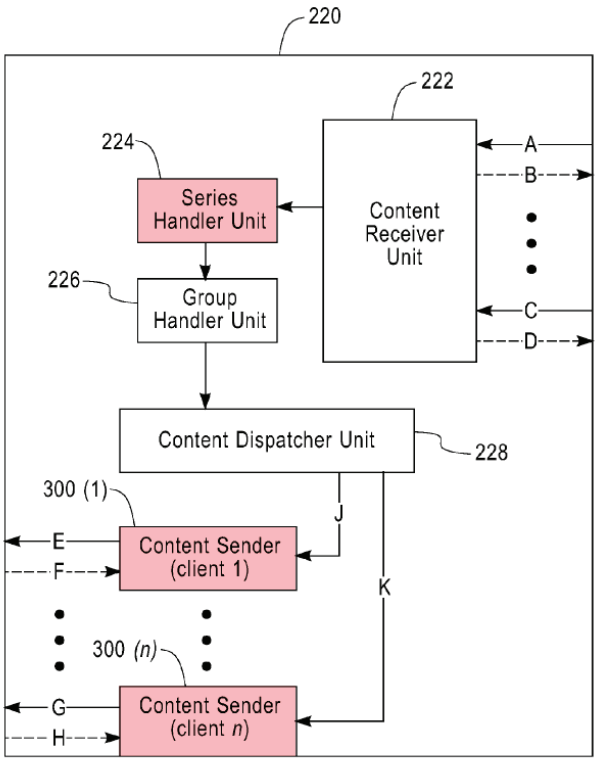


FIG. 3

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, ¶414.

Munson's Figure 5 illustrates a structure for queuing messages in Content Sender (300). Traynor, ¶414.

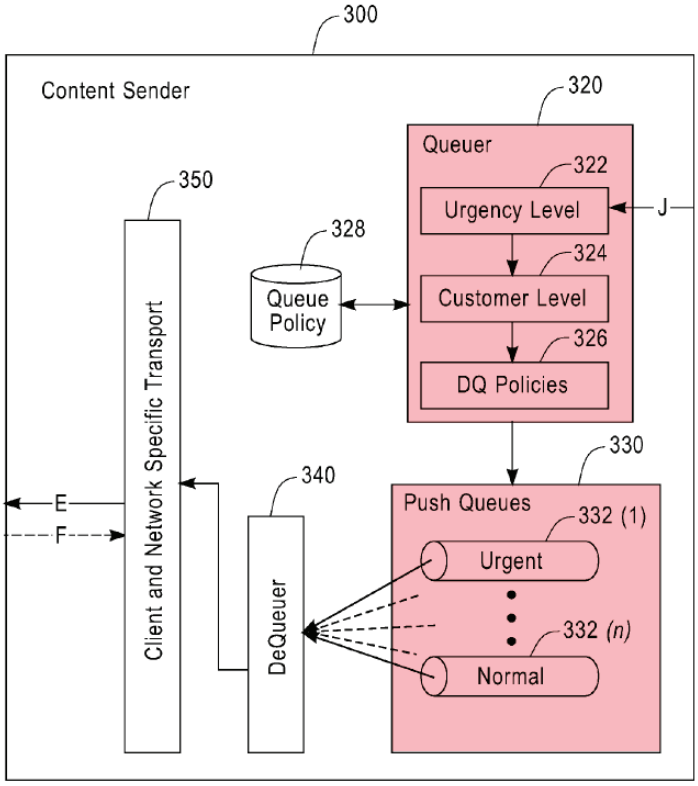


FIG. 5

Munson, Fig. 5.

POSITAs would have found it obvious to implement computer memory and programming to deliver messages from memory, consistent with Munson, including upon occurrence of certain trigger events. §VI.H.3; Traynor, ¶415. Thus, in the Houghton-Munson 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 a given one of the wireless end-user* devices). Traynor, ¶¶407-17.

g. 1[d3]/15[d2]

Houghton-Munson teaches 1[d3]/15[d2]. Traynor, ¶¶418-30.

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, ¶¶419-421.

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 '192Pat, 38:50-63. §VI.A.2.g (Ground 1A, 1[d3]/15[d2]); Traynor, ¶¶422-23. Houghton's trigger event can be an alarm or notification received from another device (as described above). Traynor, ¶¶420, 423. Message delivery can be adjusted based on whether the message to be delivered is time-critical or "non-time-critical." Houghton, 23-24; Traynor, ¶424.

Houghton's push client also sends a message based on a user request to push server, requesting message delivery. Traynor, ¶426. "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, ¶426. "[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, ¶425. Houghton-Munson teaches this limitation because Houghton contemplates configuring the system for "any trigger event," including those described in Munson as additional example triggers in a similar networking system. Houghton, 21; Traynor, ¶425.

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, ¶424.

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, ¶¶418-30.

h. 1[d4]/15[d3]

Houghton-Munson teaches 1[d4]/15[d3]. Traynor, ¶¶431-35.

Houghton's push server uses a stack of secure protocols for connections between push server/push client (*transport-services stack*; §VI.H.4.b (1[a]/15[a])) to establish a connection between mobile terminals and push server (*respective secure-message links*; §VI.H.4.b (1[a]/15[a]). §VI.H.4.c (1[b]/15[b]); Traynor, ¶432. Houghton-Munson stores/buffers received push messages in push server's memory, (§VI.H.4.f ([1d1-d2]/[15d1]); Traynor, ¶432), 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 (§VI.H.4.g (1[d3]/15[d2]); Traynor, ¶433).

In Houghton, "intelligence" "route[s] messages received on behalf of the wireless terminal to and from the server." Houghton, 14. Such message delivery happens using the secure-message link established/maintained between the push server and the terminal's push client. Houghton, cl. 1 ("push server sends, in

response to predetermined trigger events," application data to "a push application" using a "client-initiated" permanent connection); Traynor, ¶434. Upon applying Munson's teachings regarding a memory/buffer (§VI.H.4.f ([1d1-d2]/[15d1])), POSITAs would have appreciated Munson's content senders operate similarly to Houghton's routing to transmit messages from the buffer to the client device. §VI.H.3 (motivation); Munson, [0036]-[0037]; Traynor, ¶432.

Accordingly, Houghton-Munson teaches push server (*message link 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 (*the given one of the wireless end-user devices*). Traynor, ¶¶431-35.

5. Claim 5

a. 5[a]-5[b]

Houghton-Munson teaches 5[a]-5[b]. Traynor, ¶¶436-40.

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, ¶438. 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 server uses a stack of secure protocols (*transport-services stack*; §VI.H.4.b (1[a]/15[a])) for connections between push server and push client and may encrypt those connections (*respective secure-message links*; §VI.H.4.b (1[a]/15[a])), POSITAs would have understood/found obvious those protocol-based connections receive messages on push server, including data (*upload messages*) forwarded by the push client(s) (*respective device-link agents*) from some of applications 805 (*device software components*). Traynor, ¶¶439-40.

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 (*device respective software component*) requested delivery. Traynor, ¶¶400, 447 (citing Lee, [0011], [0022], cl.1).

b. 5[c][12]

Houghton-Munson teaches 5[c]. Traynor, ¶¶441-443.

Houghton's push server (*network-server system*) includes interfaces facilitating communications between applications/application servers, push servers,

[12] *See* n.9; Traynor, ¶442 n.7.

and push clients to deliver messages and associated data to one or more application servers. §VI.H.5.a ([5a-b]); Traynor, ¶441. Moreover, the push server uses an API/IP connection for communications between the push server and the application server (*the network server system using the interface to a network*), and POSITAs would have understood this interface/network connection would have been used for communicating upload messages to the application server(s) from the push server (*to deliver content from the upload messages to the respective identified network elements*). §VI.H.5.a ([5a-b]); Traynor, ¶442.

6. Claim 6

Houghton-Munson teaches claim 6. Traynor, ¶¶444-49.

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. §§VI.H.4.c (1[b]/15[b]), VI.H.4.e (1[c2]/15[c2]); Traynor, ¶¶400, 446 (citing Lee, [0013], [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" occuring across applications when receiving a push message. Houghton, 21; Traynor, ¶¶400, 447

(citing Lee, [0022]-[0023]). 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, ¶447.

7. Claim 7

Houghton-Munson teaches claim 7. Traynor, ¶¶450-53.

Houghton's terminal includes "software" that "initiates and maintains contact with the server using Internet technologies" and this "mobile-initiated permanent IP connection allows the server to…push messages…to the mobile terminal." Houghton, 11, claim 1; Traynor, ¶450. Moreover, Houghton uses SSL for secure messaging between push client and push server, and it was well-known for SSL communications to be client device-initiated. §VI.H.4.b (1[a]/15[a]), Traynor, ¶451 (citing Lu, 29:31-30:24).

Accordingly, Houghton-Munson teaches the mobile terminal's software (*the device messaging agent on at least one of the wireless end-user devices*) initiates contact with the push server's TCP/IP transport protocols with SSL using the Internet (*initiates the respective secure Internet data-message link to the transport-services stack*). Traynor, ¶¶450-53.

8. Claim 9

Houghton-Munson teaches claim 9. Traynor, ¶¶454-58.

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, ¶¶455-56.

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, ¶456; §VI.H.4.g (1[d3]/15[d2]).

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, ¶457. 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.*

9. Claim 10

Houghton-Munson teaches claim 10. Traynor, ¶¶459-62.

Houghton's push client sends a "periodic message" so the connection does not expire. §VI.H.8 (cl. 9); 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, ¶460.

POSITAs would have understood/found obvious the period of the regularly-repeating timer for triggering message delivery in Houghton-Munson (*see* §VI.H.8 (cl. 9)) would be some degree shorter (*fractionally shorter*) than this *maximum data message interval beyond which the secure message link is taken down*. Traynor, ¶461. 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.*

10. Claims 11-12

Houghton-Munson teaches claims 11-12. Traynor, ¶¶463-68.

In Houghton, "client-side events" result in "pushing application commands to a mobile terminal." Houghton, 14, 16, 21, Fig. 4; Traynor, ¶464. Trigger events

include "messaging actions" or a "notification." Houghton, 14, 21. Houghton's "client" notifies the "server" of a "terminal event" triggering 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, ¶466.

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. §VI.H.8 (cl.9); Houghton, 26; Traynor, ¶467; '192Pat, 38:50-63.

Accordingly, 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 message link from the device link agent*, including a *heartbeat message* or a *request received from the given device link agent*). Traynor, ¶¶464-68.

11. Claim 13

Houghton-Munson teaches claim 13. Traynor, ¶¶469-70.

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 network-element message*) from an application server (*network element*). Traynor, ¶469.

I. Ground 2B: Houghton-Munson-TS-23.140 (Claims 1, 5-7, 9-13, 15)

If Houghton-Munson does not render obvious claims 1, 5-7, 9-13, 15—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, ¶471.

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, ¶472.

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, ¶473. POSITAs would have had a reasonable expectation of success in doing so because it merely involved 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. TS-23.140, 54; Traynor, ¶473.

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 those messages are directed to the intended application or identify an application to another network element. Traynor, ¶474. 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, ¶475.

J. Ground 2C: Houghton-Munson-Shen (Claims 2, 3, 12)

1. Claim 2

Houghton-Munson-Shen teaches claim 2. Traynor, ¶¶476-87.

Houghton supplies messages through secure push-server protocols (e.g., SSL/HTTPS) (*transport-services stack*) for delivery over a secure connection (*secure-message link*) between push server (*message-link server*) and a mobile terminal's push client (*device-link agent on a wireless end-user device*). §§VI.H.4.b (1[a]/15[a]), VI.H.4.h (1[d4]/15[d3]); Traynor, ¶477. Although Houghton-Munson teaches encrypting the connection, it does not explicitly disclose encrypting the *messages*. *Id.*

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 (*encryption function to encrypt one or more messages*) for "encrypting MMS messages." §VI.B.1 (Shen); Shen, [0004], [0029]-[0034], [0059], Fig. 5; Traynor, ¶¶478-83, 486. Shen's messages can be "encode[d] and decode[d] to protect the privacy of these messages." Shen, [0054]; Traynor, ¶111.

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. §VI.B.2. (Ground 1B, cl.2); Traynor, ¶¶483-486.

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, ¶484.

Accordingly, Houghton-Munson-Shen teaches providing an encrypt function to encrypt messages supplied to the TCP/IP transport protocols with SSL implemented by push server (*transport-services stack*) for delivery over a secure connection (*secure-message link*) between push server (*message-link server*) and a mobile terminal's push client (*device-link agent on a wireless end-user device*). Traynor, ¶¶476-87.

2. Claim 3

Houghton-Munson-Shen teaches claim 3. Traynor, ¶¶488-90.

Houghton-Munson-Shen teaches *encrypting messages* and transporting the encrypted messages to a particular push client (*device-link agent*) on a mobile terminal (*wireless end-user device*). §VI.J.1 (cl. 2); Traynor, ¶489. POSITAs would have further understood based on Houghton that the encrypted messages would be

transported by push server using TCP/IP transport protocols with SSL/HTTPS encryption (*encryption on the transport services stack*). §VI.H.4.b (1[a]/15[a]); Houghton, cl. 10; Traynor, ¶489.

3. Claim 12

Houghton-Munson-Shen teaches claim 12. Traynor, ¶¶491-95.

Houghton-Munson-Shen teaches a heartbeat message as a trigger. Traynor, ¶491. Houghton uses a "periodic message" as a message-delivery trigger to keep a connection alive (*see* §VI.H.8 (cl.9)), and monitors for when client (405) has become disconnected (Houghton, 19-20, 26-27; Traynor, ¶492).

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, ¶493. 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, ¶493. POSITA's would have understood a *heartbeat* provides a low-bandwidth and reliable mechanism for triggering message delivery. Traynor, ¶493 (citing Ex-1043, [0062]).

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, ¶494. POSITAs would have had a reasonable expectation of success doing so because Houghton and Shen already provide for similar signaling. Shen, [0034]; Traynor, ¶494.

Accordingly, Houghton-Munson-Shen teaches a message-delivery trigger comprising *the transmission of a heartbeat message* like Shen's, generated by push client (*device-link agent*). Traynor, ¶¶491-95.

K. Ground 2D: Houghton-Munson-Ellison (Claim 4)

Houghton-Munson-Ellison teaches claim 4. Traynor, ¶¶496-512.

Houghton-Munson teaches push client (*device-link agent*) and other applications executing on a mobile terminal (*one of the devices*) (Houghton, 16, 21, Fig. 4), but does not explicitly teach implementing push client in a secure environment. Traynor, ¶497.

Ellison implements software in a secure environment to protect it from potential attacks. §VI.C (Ground 1C); Traynor, ¶¶498-500.

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 the push server for the same reasons described regarding TS-23.140-Ellison. §VI.C; Traynor, ¶¶501-06.

POSITAs would have had a reasonable expectation of success in making this combination 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, ¶¶506-11 (citing Ex-1033).

Accordingly, 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 (*software components*) execute in either a separate "isolated" environment or "normal" execution environment (*outside of the secure-execution environment*). Traynor, ¶¶496-512.

L. Ground 2E: Houghton-Munson-Rakic (Claim 8)

Houghton-Munson-Rakic teaches claim 8. Traynor, ¶¶513-25.

Houghton's push clients operate on mobile terminals (*wireless end-user devices*) to receive messages over a secure connection (*secure-message link*) via a push-server (*message-link server*), including to/from various applications. Houghton, 19-21; §§VI.H.4.c (1[b]/15[b]), VI.H.4.e (1[c2]/15[c2]); Traynor, ¶514. Houghton-Munson does not explicitly provide a mechanism to verify the sender's identity or

that the destination application is the intended recipient, thus leaving the push client exposed to potential attacks. Traynor, ¶515.

Rakic recognized the risk of unverified transmissions, and proposed the use of electronic signatures to verify the parties to the transmission. §VI.D (Ground 1D). Traynor, ¶¶516-17.

POSITAs would have been motivated to implement a secure-push-message server and signature generation like Rakic within Houghton's push server with a reasonable expectation of success for the same reasons described regarding TS-23.140-Rakic. §VI.D; Traynor, ¶¶518-20.

POSITAs would have had a reasonable expectation of success in making this combination because Houghton and Rakic contemplate similar messaging architectures/systems, and POSITAs would have found it straightforward to implement Rakic's teachings within Houghton's push server. *Id.* Additionally, the resulting system's elements would perform functions performed prior to combination—Houghton's push server would communicate messages to push clients and other network elements, and Rakic's teachings (in combination) would provide secure signatures (with messages) to the receiving device(s). Traynor, ¶¶521-22.

Accordingly, Houghton-Munson-Rakic teaches a secure-push-message server (*secure server*) and electronic signatures (*secure authorization signatures*) to

indicate to a device executing Houghton's push client (*wireless end-user device*) that an authorized/registered application (*authorized software component*) is allowed to receive messages over the secure connection (*secure-message link*) via Houghton's push server (*message-link server*). Traynor, ¶¶513-25.

M. Ground 2F: Houghton-Munson-Adamczyk (Claims 9-10)

1. Claim 9

Houghton-Munson-Adamcyzk teaches claim 9. Traynor, ¶¶526-31.

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, ¶527.

POSITAs would have found it obvious to supplement Houghton-Munson's triggers (§VI.H.4.g (1[d3]/15[d2])) with Adamcyzk's additional trigger types. Traynor ¶528. 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*). *Id.* POSITAs would have been motivated to do so with a reasonable expectation of success for the same reasons described regarding TS-23.140-Adamcyzk. §VI.E; Traynor, ¶¶529-30.

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, ¶531. 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. *Id.*

2. Claim 10

Houghton-Munson-Adamcyzk teaches claim 10. Traynor, ¶532.

POSITAs would have understood/found obvious that the period of the periodic timer trigger in Adamcyzk (§VI.M.1 (cl. 9, Ground 2F)) would be *fractionally shorter* than the period for sending the "periodic message" that prevents connection expiration in Houghton-Munson—i.e., the period that sets the *maximum data message interval beyond which the secure message link is taken down*—for the same reasons described regarding Houghton-Munson. §VI.H.9 (cl.10, Ground 2A); Traynor, ¶532.

N. Ground 2G: Houghton-Munson-Gellens (Claim 14)

Houghton-Munson-Gellens teaches claim 14. Traynor, ¶¶533-38.

Houghton-Munson uses triggers (*message-delivery triggers*) to control when messages are delivered to a push client. §VI.H.4.g (1[d3]/15[d2]). Traynor, ¶534.

Houghton-Munson does not explicitly teach triggering message delivery based on an amount of data consumed by the user terminal. *Id.*

Gellens, however, recognized a need to prioritize which applications may initiate connections between a server and client, and trigger delivery of low-priority messages only after a high-priority application initiates a connection and transmits its message. §VI.G (Ground 1G). Traynor, ¶¶535-36.

For the same reasons described regarding TS-23.140-Gellens, POSITAs would have been motivated to combine Houghton-Munson and Gellens with a reasonable expectation of success to trigger transmission of low-priority application messages only after a high-priority application opens a connection and is consuming data to transmit its message (*one of the message delivery triggers is based on an amount of wireless network data usage consumed by the given one of the wireless end-user devices*). §VI.G; Traynor, ¶¶537-38.

POSITAs would have had a reasonable expectation of success in making this combination because it merely involves configuring the applications Houghton-Munson already envisions being incorporated into the message network to be assigned priorities, and control what priorities are required to establish a connection between the push server and push client. Traynor, ¶¶533-38.

VII. CONCLUSION

IPR of the Challenged Claims is respectfully requested.

VIII. 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 '192Pat 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).
- *Samsung Elecs., Ltd. v. Headwater Research LLC*, IPR2024-00010 (PTAB).

- *Headwater Research LLC v. Samsung Elecs. Co. et al.*, No. 2-23-cv-00103 (EDTX).

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. 9,615,192, 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 13,099. 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. 9,615,192 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