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; IPR2026-00106
U.S. Patent Nos. 9,615,192; 10,321,320

_____

DECLARATION OF DR. PATRICK TRAYNOR

TABLE OF CONTENTS

I. QUALIFICATIONS AND BACKGROUND 1

II. LEGAL PRINCIPLES 9

A. Anticipation 9

B. Obviousness 9

III. OVERVIEW OF CONCLUSIONS FORMED 11

A. '192 Patent 11

B. '320 Patent 13

IV. BACKGROUND KNOWLEDGE OF ONE OF ORDINARY SKILL IN THE ART 15

V. TECHNOLOGY BACKGROUND 16

VI. THE '192 PATENT 22

A. Overview of the '192 Patent 22

B. Prosecution History of the '192 Patent 25

C. Interpretations of the '192 Patent Claims at Issue 33

VII. THE '320 PATENT 34

A. Overview of the '320 Patent 34

B. Prosecution History of the '320 Patent 35

C. Interpretations of the '320 Patent Claims at Issue 35

VIII. OVERVIEW OF THE PRIOR ART 36

A. TS-23.140 (Ex-1004) 36

B. Shen (Ex-1014) 39

C. Rakic (Ex-1015) 43

D. Houghton (Ex-1007) 46

E. Munson (Ex-1017) 49

F. Ellison (Ex-1019) 50

G. Adamczyk (Ex-1021) 52

H. Herzog (Ex-1008) 53

I. Gellens (Ex-1020) 54

J. Pazhyannur (Ex-1037) 55
K. Fok (Ex-1038) 56
IX. ANALYSIS OF THE '192 PATENT 57
A. Ground 1A: Claims 1, 5, 6, 7, 9, 11-13, and 15 Are Rendered Obvious By TS-23.140 57
1. Claims 1 and 15 57
a. 1[pre]/15[pre] 57
b. 1[a]/15[a] 59
c. 1[b]/15[b] 68
d. 1[c1]/15[c1] 75
e. 1[c2]/15[c2] 78
f. 1[d1]-1[d2]/15[d1] 82
g. 1[d3]/15[d2] 85
h. 1[d4]/15[d3] 93
2. Claim 5 94
a. 5[a] 95
b. 5[b] 97
c. 5[c] 100
3. Claim 6 101
4. Claim 7 103
5. Claim 9 105
6. Claims 11-12 106
7. Claim 13 107
B. Ground 1B: Claims 2, 3, and 12 Are Rendered Obvious By TS-23.140 and Shen 108
1. Claim 2 108
2. Claim 3 112
3. Claim 12 113
C. Ground 1C: Claim 4 Is Rendered Obvious By TS-23.140 and Ellison 114

1. Claim 4 114

D. Ground 1D: Claim 8 Is Rendered Obvious By TS-23.140 and Rakic 124

1. Claim 8 125

E. Ground 1E: Claim 9 Is Rendered Obvious By TS-23.140 and Adamczyk 131

1. Claim 9 131

F. Ground 1F: Claim 10 Is Rendered Obvious By TS-23.140 and Herzog, or TS-23.140, Adamczyk and Herzog 134

1. Claim 10 134

G. Ground 1G: Claim 14 Is Rendered Obvious by TS-23.140 and Gellens 137

1. Claim 14 137

H. Ground 2A: Claims 1, 5-7, 9-13, and 15 are Rendered Obvious by Houghton and Munson 139

1. Combination of Houghton and Munson 139

2. Claim Analysis 145

a. Claims 1 and 15 145

i. 1[pre]/15[pre] 145

ii. 1[a]/15[a] 147

iii. 1[b]/15[b] 157

iv. 1[c1]/15[c1] 166

v. 1[c2]/15[c2] 170

vi. 1[d1]-1[d2]/15[d1] 176

vii. 1[d3]/15[d2] 181

viii. 1[d4]/15[d3] 187

b. Claim 5 189

i. 5[a]-[b] 189

ii. 5[c] 191

c. Claim 6 192

d. Claim 7.....193
e. Claim 9.....195
f. Claim 10.....197
g. Claims 11-12.....199
h. Claim 13.....201
I. Ground 2B: Houghton, Munson, and TS-23.140 (Claims 1, 15).....201
J. Ground 2C: Claims 2, 3, and 12 Are Rendered Obvious by Houghton, Munson, and Shen 204
1. Claim 2.....204
2. Claim 3.....209
3. Claim 12.....210
K. Ground 2D: Claim 4 Is Rendered Obvious by Houghton, Munson, and Ellison.....211
1. Claim 4.....211
L. Ground 2E: Claim 8 Is Rendered Obvious by Houghton, Munson, and Rakic.....220
1. Claim 8.....220
M. Ground 2F: Claims 9 and 10 Are Rendered Obvious by Houghton, Munson, and Adamczyk.....225
1. Claim 9.....225
2. Claim 10.....227
N. Ground 2G: Claim 14 Is Rendered Obvious by Houghton, Munson, and Gellens.....228
1. Claim 14.....228
X. ANALYSIS OF THE '320 PATENT 230
A. Ground 1A: Claims 1-3, 5-7, 10-11, 15, and 18 Are Rendered Obvious by TS-23.140 230
1. Claim 1.....231
a. 1[pre]-1[a].....231
b. 1[b1].....233
c. 1[b2].....241

d. 1[c1] 248
e. 1[c2] 252
f. 1[d1]-[d2] 255
g. 1[d3] 258
h. 1[d4] 266
i. 1[e1] 268
j. 1[e2] 269
k. 1[e3] 270
l. 1[e4]. 273
2. Claim 2 277
3. Claim 3 278
4. Claim 5 279
5. Claim 6 280
6. Claim 7 280
7. Claim 10 281
8. Claim 11 282
9. Claim 15 286
a. 15[a]-15[b] 286
b. 15[c] 291
10. Claim 18 293
B. Ground 1B: Claim 3 Is Rendered Obvious by TS-23.140 and Adamczyk 295
1. Claim 3 295
C. Ground 1C: Claim 4 is Rendered Obvious by TS-23.140 and Herzog, With or Without Adamczyk 298
1. Claim 4 298
D. Ground 1D: Claims 6, 12, 13, and 16 are Rendered Obvious by TS-23.140 and Shen 301
1. Claim 6 301
2. Claim 12 302

3. Claim 13 307
4. Claim 16 308
E. Ground 1E: Claims 8 and 9 are Rendered Obvious by TS-23.140 and Pazhyannur 310
1. Claims 8-9 310
F. Ground 1F: Claim 14 is Rendered Obvious by TS-23.140 and Ellison 312
1. Claim 14 312
G. Ground 1G: Claim 17 is Rendered Obvious by TS-23.140 and Fok 322
1. Claim 17 322
H. Ground 2A: Claims 1-7, 10-11, 15, and 18 Are Rendered Obvious by Houghton and Munson 326
1. Combination of Houghton and Munson 326
2. Claim Analysis 332
a. Claim 1 332
i. 1[pre]-1[a] 331
ii. 1[b1] 333
iii. 1[b2] 342
iv. 1[c1] 351
v. 1[c2] 355
vi. 1[d1]-[d2] 358
vii. 1[d3] 363
viii. 1[d4] 368
ix. 1[e1] 370
x. 1[e2] 370
xi. 1[e3] 371
xii. 1[e4] 376
b. Claim 2 378
c. Claim 3 379

d. Claim 4....382
e. Claim 5-6....383
f. Claim 7....385
g. Claim 10....386
h. Claim 11....387
i. Claim 15....388
i. 15[a]-15[b]....387
ii. 15[c]....388
j. Claim 18....390
I. Ground 2B: Claim 1 Is Alternatively Rendered Obvious by Houghton, Munson, and TS-23.140....392
J. Ground 2C: Claims 3 and 4 Are Rendered Obvious by Houghton, Munson, and Adamczyk....394
1. Claim 3....395
2. Claim 4....396
K. Ground 2D: Claims 6, 12, 13, and 16 Are Rendered Obvious by Houghton, Munson, and Shen....396
1. Claim 6....396
2. Claim 12....398
3. Claim 13....403
4. Claim 16....404
L. Ground 2E: Claims 8 and 9 Are Rendered Obvious by Houghton, Munson, and Pazhyannur....405
1. Claims 8-9.....405
M. Ground 2F: Claim 14 Is Rendered Obvious by Houghton, Munson, and Ellison....408
1. Claim 14....408
N. Ground 2G: Claim 17 Is Rendered Obvious by Houghton, Munson, and Fok....416
1. Claim 17....416
XI. ADDITIONAL REMARKS....421

I, Patrick Gerard Traynor, of Gainesville, Florida, declare that:

I. QUALIFICATIONS AND BACKGROUND

1. My name is Patrick Gerard Traynor and I have been retained as an expert witness by Amazon in the matter of *Headwater Research LLC v. Amazon.com Services LLC and Amazon Web Services, Inc.* My qualifications for forming these conclusions are summarized below and are additionally included in my curriculum vitae provided herewith.

2. I earned a B.S. in Computer Science from the University of Richmond in 2002 and an M.S. and Ph.D. in Computer Science and Engineering from the Pennsylvania State University in 2004 and 2008, respectively. My dissertation, entitled "Characterizing the Impact of Rigidity on the Security of Cellular Telecommunications Networks," focused on security problems that arise in cellular infrastructure when gateways to the broader Internet were created.

3. I am currently a Professor in the Department of Computer and Information Science and Engineering (CISE) at the University of Florida. I was hired under the "Rise to Preeminence" Hiring Campaign and serve as the Interim Chair for my Department. I also hold the endowed position of the John and Mary Lou Dasburg Preeminent Chair in Engineering.

4. Prior to joining the University of Florida, I was an Associate Professor from March to August 2014 and an Assistant Professor of Computer Science from

2008 to March 2014 at the Georgia Institute of Technology. I have supervised many Ph.D., M.S., and undergraduate students during the course of my career.

5. My area of expertise is security, especially as it applies to mobile systems and networks, including cellular networks and messaging. Also, as noted above, my dissertation concentrated on security issues in messaging for cellular systems. I regularly teach students taking my courses and participating in my research group to program and evaluate software and architectures for mobile and cellular systems. I have taught courses on the topics of network and systems security, cellular networks, and mobile systems at both Georgia Tech and the University of Florida. I also advised and instructed the Information Assurance Officer Training Program for the United States Army Signal Corps in the Spring of 2010.

6. I have received numerous awards for research and teaching, including being named a Kavli Fellow (2017), a Fellow of the Center for Financial Inclusion (2016), and a Research Fellow of the Alfred P. Sloan Foundation (2014). I also won the Lockheed Inspirational Young Faculty Award (2012), was awarded a National Science Foundation (NSF) CAREER Award (2010), and received the Center for Enhancement of Teaching and Learning at Georgia Tech's "Thanks for Being a Great Teacher" Award (2009, 2012, 2013).

7. I have published over 100 articles in top conferences and journals in the areas of information security, mobile systems, and networking. Many of my results

are highly cited, and I have received multiple "Best Paper" Awards. I have also written a book entitled "Security for Telecommunications Networks", which is used in wireless and cellular security courses at a number of top universities.

8. I am a Senior Member of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). I am also a member of the USENIX Advanced Computing Systems Association.

9. I served as an Associate Editor for IEEE Security and Privacy Magazine, and currently serve as an Associate Editor for ACM Transactions on AI Security and Privacy (TAISP). I have also been the Program Chair for eight conferences and workshops, and have served as a member of the Program Committee for over 50 different conferences and workshops. I also previously served as the Security Subcommittee Chair for the ACM US Technology Policy Committee (USACM).

10. I was a co-Founder and Research Fellow for a private start-up, Pindrop Security, from 2012 to 2014. Pindrop provides anti-fraud and authentication solutions for Caller-ID spoofing attacks in enterprise call centers by creating and matching acoustic fingerprints. Pindrop Security currently employs over 200 people, and their technology is based off of my research (U.S. Patent 9,037,113 B2).

11. I was a co-Founder and Chief Executive of a private start-up, CryptoDrop. CryptoDrop developed a ransomware detection and recovery tool to

provide state of the art protection to home, small business, and enterprise users. This technology was also based off of my research (U.S. Patent 10,685,114 B2).

12. I was also a co-Founder and Chief Executive of a private start-up, Skim Reaper. Skim Reaper developed tools to detect credit card skimming devices, and worked with a range of banks, international law enforcement, regulators, and retailers. This technology was also based off of my research (U.S. Patent 10,496,914 B2).

13. I am a named inventor on over ten US patents. These patents generally describe methods for determining the origin and path taken by phone calls as they traverse various networks, cryptographically authenticating phone calls, providing a secure means of indoor localization using mobile/wireless devices, detecting credit card skimmers, identifying cloned credit cards, blocking ransomware from encrypting data, and more.

14. My curriculum vitae, included with this declaration, includes a list of publications on which I am a named author. It contains further details regarding my experience, education, publications, and other qualifications to render an expert opinion in connection with this proceeding.

15. In writing this Declaration, I have considered the following: my own knowledge and experience, including my work experience in mobile systems and networks; my experience in teaching those subjects; and my experience in working

with others involved in those fields. In addition, I have analyzed the following publications and materials, in addition to other materials I cite in my declaration:

- U.S. Patent No. 9,615,192 ('192 Patent, Ex-1001), and its accompanying prosecution history ('192 Patent, Ex-1002)
- U.S. Patent No. 10,321,320 ('320 Patent, Ex-1001), and its accompanying prosecution history ('320 Patent, Ex-1002)
- 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") (Ex-1004)
- U.S. Patent Pub. No. 2006/0190720 to Ozaki et al. ("Ozaki") (Ex-1005)
- WO 2008/048075 to Lee et al. ("Lee") (Ex-1006)
- WO 2006/077283 to Houghton et al ("Houghton") (Ex-1007)
- U.S. Patent Pub. No. U.S. 2009/0158397 to Herzog et al. ("Herzog") (Ex-1008)
- U.S. Patent No. 7,925,717 to Chou ("Chou") (Ex-1009)
- Open Mobile Alliance; Multimedia Messaging Service Architecture Overview (MMSARCH) specification*, available at* *https://www.openmobilealliance.org/release/MMS/V1_1-20040715-A/OMA-WAP-MMS-ARCH-V1_1-20040715-A.pdf* (Ex-1010)

- Open Mobile Alliance; OMA-ERELD-MMS-v1_2-20030923-C, Enabler Release Definition for MMS Version 1.2," available at https://www.openmobilealliance.org/release/MMS/V1_2-20030923-C/OMA-ERELD-MMS-V1_2-20030923-C.pdf (Ex-1011)
- U.S. Patent No. 7,509,487 to Lu et al. ("Lu") (Ex-1012)
- Technical Specification Group Services and System Aspects Meeting #19, TSGS#19(03)0167, European Telecommunications Standards Institute, 3ETSI / AT-F TD18 – Rapporteur Meeting (4-6 Feb. 2003), available at https://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_19/Docs/PDF/SP-030167.pdf (Ex-1013)
- U.S. Patent Pub. No. 2005/0207379 to Shen et al. ("Shen") (Ex-1014)
- U.S. Patent Pub. No. 2009/0282256 to Rakic et al. ("Rakic") (Ex-1015)
- U.S. Patent Publication No. 2009/0240807 to Munson et al ("Munson") (Ex-1017)
- EP Patent Application EP1853044A1 to Shenfield ("Shenfield") (Ex-1018)
- U.S. Patent No. 7,082,615 B1 ("Ellison") (Ex-1019)
- U.S. Patent Pub. No. 2005/0144294 to Gellens et al. ("Gellens") (Ex-1020)
- U.S. Patent Pub. No. 2008/0162637 to Adamcyzk et al. ("Adamcyzk") (Ex-1021)

- IETF RFC 793, Transmission Control Protocol (Sept. 1981), available at https://www.ietf.org/rfc/rfc793.txt (Ex-1027)
- The TLS Protocol Version v 1.0 (Jan. 1999), available at https://datatracker.ietf.org/doc/html/rfc2246 (Ex-1028)
- Needham et al., "Using Encryption for Authentication in Large Networks of Computers" (ACM, Vol. 21, No. 12, Dec. 1978) ("Needham") (Ex-1030)
- Schroeder et al., "A Hardware Architecture for Implementing Protection Rings" (ACM, Vol. 15, No. 3, Mar. 1972) ("Schroeder") (Ex-1031)
- Saltzer et al., "The Protection of Information in Computer Systems" (IEEE Proceedings, Vol. 63, No. 9, Sept. 1975) ("Saltzer") (Ex-1032)
- Li et al., "Symbian OS platform security model" (Login Magazine, Aug. 2010) (Ex-1033)
- 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.tech-insider.org/free-software/research/acrobat/910605.pdf ("Zimmerman") (Ex-1034)
- B. Ramsdell, S/MIME Version 3 Message Specification, IETF RFC 2633, June 1999, available at https://datatracker.ietf.org/doc/html/rfc2633 ("Ramsdell") (Ex-1035)

- Mostafa, "Transporting data between wireless applications using a messaging system—MMS" (Wireless Comms. and Mobile Computing, July 7, 2006) ("Mostafa") (Ex-1036)
- U.S. Patent Pub. No. 2008/0243999 to Pazhyannur et al. ("Pazhyannur") (Ex-1037)
- U.S. Patent Pub. No. 2008/0215883 to Fok et al. ("Fok") (Ex-1038)
- U.S. Patent Pub. No. 2003/0126282 to Sarkar et al. ("Sarkar") (Ex-1039)
- U.S. Patent Pub. No. 2007/0037610 to Logan ("Logan") (Ex-1040)
- U.S. Patent Pub. No. 2005/0091380 to Gonen et al. ("Gonen") (Ex-1041)
- U.S. Patent Pub. No. 2004/0258063 to Raith et al. ("Raith") (Ex-1042)
- U.S. Patent Pub. No. 2004/0085894 to Wang et al. ("Wang") (Ex-1043)
- U.S. Patent No. 6,094,424 to Kalmanek et al. ("Kalmanek") (Ex-1044)
- 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") (Ex-1045)
- U.S. Patent Pub. No. 2003/0193967 to Fenton et al. ("Fenton") (Ex-1046)
- U.S. Patent Pub. No. 2004/011147 to Trossen et al. ("Trossen") (Ex-1047)

- U.S. Patent Pub. No. 2005/028253 to Andreasson et al. ("Andreasson") (Ex-1048)

II. LEGAL PRINCIPLES

A. Anticipation

16. I have been informed that a patent claim is invalid as anticipated under 35 U.S.C. § 102 if each and every element of a claim, as properly construed, is found either explicitly or inherently in a single prior art reference. Under the principles of inherency, if the prior art necessarily functions in accordance with, or includes the claimed limitations, it anticipates.

17. I have been informed that a claim is invalid under 35 U.S.C. § 102(a) if the claimed invention was known or used by others in the U.S., or was patented or published anywhere, before the applicant's invention. I further have been informed that a claim is invalid under 35 U.S.C. § 102(b) if the invention was patented or published anywhere, or was in public use, on sale, or offered for sale in this country, more than one year prior to the filing date of the patent application (critical date). And a claim is invalid, as I have been informed, under 35 U.S.C. § 102(e), if an invention described by that claim was described in a U.S. patent granted on an application for a patent by another that was filed in the U.S. before the date of invention for such a claim.

B. Obviousness

18. I have been informed that a patent claim is invalid as "obvious" under 35 U.S.C. § 103 in light of one or more prior art references if it would have been obvious to a POSITA, taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so called "secondary considerations" of non-obviousness, which include: (i) "long felt need" for the claimed invention, (ii) commercial success attributable to the claimed invention, (iii) unexpected results of the claimed invention, and (iv) "copying" of the claimed invention by others. For purposes of my analysis above and because I know of no indication from the patent owner or others to the contrary, I have applied a date of January 28, 2009, as the date of invention in my obviousness analyses, although in many cases the same analysis would hold true even at an earlier time than January 28, 2009.

19. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason to modify the single prior art reference, or combine two or more references, in order to achieve the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference or references themselves, the knowledge or "common sense" of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole.

I have been informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

III. OVERVIEW OF CONCLUSIONS FORMED

20. This expert Declaration explains the conclusions that I have formed based on my analysis.

A. '192 Patent

21. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1, 5-7, 9, 11-13, and 15 of the '192 Patent are obvious over TS-23.140.

22. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 2, 3, and 12 of the '192 Patent are obvious over TS-23.140 in view of Shen.

23. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 4 of the '192 Patent is obvious over TS-23.140 in view of Ellison.

24. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 8 of the '192 Patent is obvious over TS-23.140 in view of Rakic.

25. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 9 of the ’192 Patent is also obvious over TS-23.140 in view of Adamczyk.

26. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 10 of the ’192 Patent is obvious over TS-23.140 in view of Herzog (with or without Adamcyzk).

27. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 14 of the ’192 Patent is obvious over TS-23.140 in view of Gellens.

28. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1, 5-7, 9-13, and 15 of the ’192 Patent are obvious over Houghton in view of Munson and further in view of TS-23.140.

29. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 2, 3, and 12 of the ’192 Patent are obvious over Houghton in view of Munson and Shen.

30. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 4 of the ’192 Patent is obvious over Houghton in view of Munson and Ellison.

31. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 8 of the ’192 Patent is obvious over Houghton in view of Munson and Rakic.

32. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 9 and 10 of the ’192 Patent are also obvious over Houghton in view of Munson and Adamczyk.

33. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 14 of the ’192 Patent is obvious over Houghton in view of Munson and Gellens.

B. ’320 Patent

34. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1-3, 5-7, 10-11, 15, and 18 of the ’320 Patent are obvious over TS-23.140.

35. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 3 of the ’320 Patent is also obvious over TS-23.140 in view of Adamczyk.

36. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 4 of the ’320 Patent is also obvious over TS-23.140 in view of Adamczyk or over TS-23.140 in view of Adamczyk and Herzog.

37. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 6, 12-13, and 16 of the ’320 Patent are obvious over TS-23.140 in view of Shen.

38. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 8 and 9 of the ’320 Patent are obvious over TS-23.140 in view of Pazhyannur.

39. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 14 of the ’320 Patent is obvious over TS-23.140 in view of Ellison.

40. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 17 of the ’320 Patent is obvious over TS-23.140 in view of Fok.

41. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1-7, 10-11, 15, and 18 of the ’320 Patent are obvious over Houghton in view of Munson and further in view of TS-23.140.

42. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 3 and 4 of the ’320 Patent are also obvious over Houghton in view of Munson and Adamczyk.

43. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 6, 12-13, and 16 of the ’320 Patent are obvious over Houghton in view of Munson and Shen.

44. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 8 and 9 of the ’320 Patent is obvious over Houghton in view of Munson and Pazhyannur.

45. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 14 of the ’320 Patent is obvious over Houghton in view of Munson and Ellison.

46. Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 17 of the ’320 Patent is also obvious over Houghton in view of Munson and Fok.

IV. BACKGROUND KNOWLEDGE OF ONE OF ORDINARY SKILL IN THE ART

47. Based on the foregoing and upon my experience in this area, a person of ordinary skill in the art (“POSITA”) relating to the subject matter of the ’320 or ’192 Patent as of January 28, 2009 (“Critical Date”) would have had (1) at least a bachelor’s degree in computer science, electrical engineering, or a related field, and (2) three to five years of experience in services and application implementation in communications networks. Additional graduate education could substitute for professional experience, and vice versa.

48. Based on my experiences, I have a good understanding of the capabilities of a POSITA as I was such an individual at the time of the Critical Date. Moreover, I have taught, participated in organizations, and worked closely with many such persons over the course of my career.

V. TECHNOLOGY BACKGROUND

49. The original and still common configuration for networked computing is the client/server architecture. As an early example, users of mainframe computers in the 1960s relied on simple "thin clients" with little to no processing power of their own to outsource their computing tasks to mainframes.

50. The benefits of the client/server architecture have long outlived the mainframe era. Client devices, generally with fewer resources (e.g., storage, bandwidth, processing capabilities, power, uptime), still often rely on servers to centralize services and for content distribution. As an example, an email or instant messaging server would not only be better provisioned than a mobile phone communicating with it, but also far more likely to be continuously powered and available on the network.

51. Because servers are better provisioned than their corresponding clients, their communication patterns are generally many-to-one. That is, because of their significant resources (and the expense in provisioning them), a small number of servers are expected to interact with a large number of clients.

52. Networking between the devices is facilitated by networking stacks in both the client and the server. The stacks implement what is known as the "Internet Model," which creates five layers of abstraction for network function: Physical Layer, Data Link Layer, Network Layer, Transport Layer, and Application Layer. The advantage to this approach is that it abstracts away most of the complexity of networking, allowing applications developers to take advantage of communications services without having to know all of the details required to provide said communications.

53. Communication over a network often requires additional security measures. For instance, an adversary may attempt to read or modify the communications between two parties (in both public and private networks). Before the Critical Date, the security community had created a wide array of mechanisms (specifically, protocols using encryption) allowing for secure communications.

54. These protocols are neither new nor uncommon. For instance, Needham and Schroeder developed the first network-based encryption key establishment protocol in 1978 (R. Needham and M. Schroeder, "Using encryption for authentication in large networks of computers." Communications of the ACM. Vol. 21, No. 12, pp. 993–99, Dec. 1978)—Ex-1030. This protocol formed the basis of security protocols (e.g., Kerberos) included in many popular operating systems from the late 1980s and beyond.

55. The need to provide security mechanisms that could be used easily across all communications quickly became widespread. Instead of requiring individual applications or processes to develop their own secure communications protocols, efforts to build secure communications layers into network stacks began in the early 1990s.

56. For instance, the Internet Engineering Task Force (IETF) formed the IP Security (IPsec) Protocol working group in 1992 (IP Security Protocol (ipsec), https://datatracker.ietf.org/wg/ipsec/history/). The IPsec protocol, which is implemented in nearly all modern operating systems, can provide cryptographic guarantees for confidentiality and integrity for all communications data above the Network Layer.

57. Another important effort to provide security services directly in the network stack was the Secure Sockets Layer (SSL) standardization process, which began in 1994 (SSL 0.2 Protocol Specification, https://www.archive.mozilla.org/projects/security/pki/nss/ssl/draft02.html, Feb. 9, 1995). This standard eventually evolved into the Transport Layer Security (TLS) protocol.

58. TLS is implemented above the Transport Layer and provides security for applications. Possibly the most visible of all security protocols, TLS is used widely to secure applications ranging from the Web (e.g., "https" in browsers) and

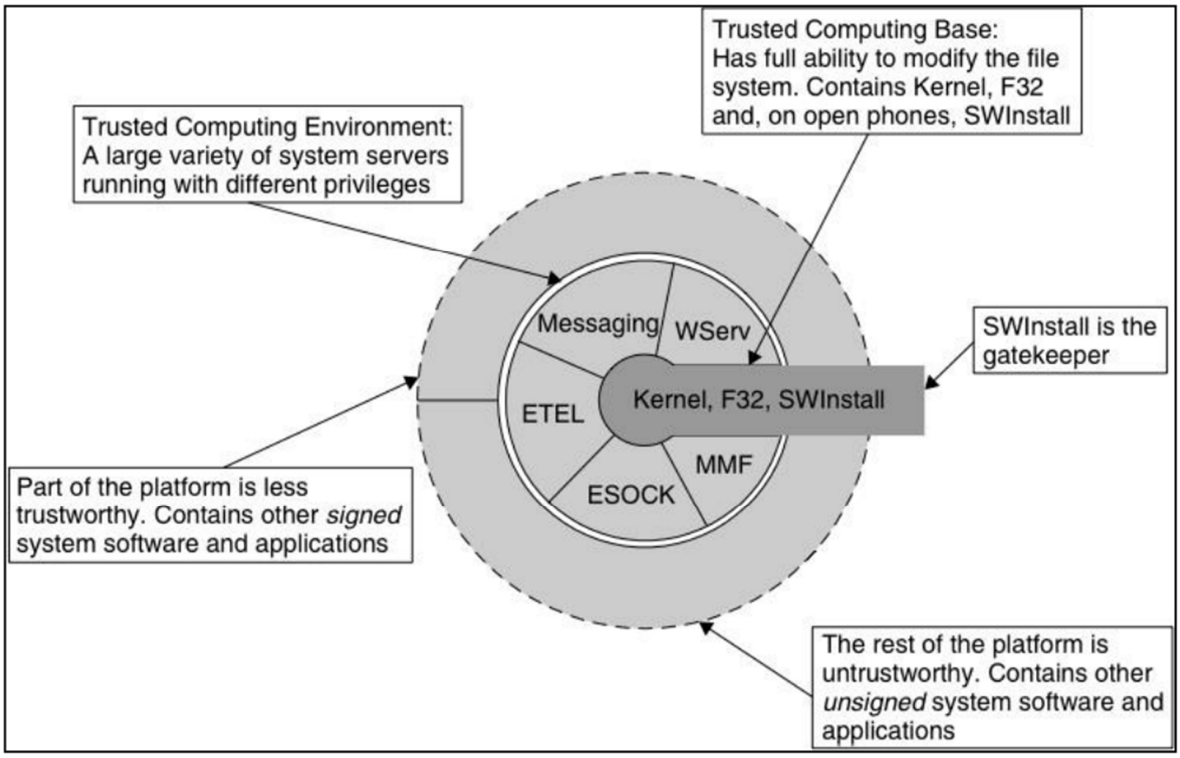
email to teleconferencing and secure messaging. TLS connections now represent over 80% of all requests on the web (Google Transparency Project - HTTPS encryption on the web, https://transparencyreport.google.com/https/overview?hl=en, Aug. 2024).

59. Security mechanisms within individual systems have also long been known to the state of the art. Separation between process (e.g., different applications, users, operating system-level processes) has long been implemented in modern operating systems. For example, classical time-sharing systems from the 1960s separated user processes into isolated time slots not only to allow for seemingly concurrent execution, but also to limit how those processes interacted with each other.

60. Many different techniques were considered. Strong protection from and isolation supported by both hardware and software were codified in the MULTICS system in 1969 (M. Schroeder and J. Saltzer, A Hardware Architecture for Implementing Protection Rings, Communications of the ACM, Vol. 15, No. 3, Mar. 1972). March 1972)—Ex-1031. Using "Protection Rings," operating systems could allow not only for the separation of different processes depending on their security needs, but also allow for communication across such security boundaries through highly controlled mechanisms. Further innovations in the community lead to increasingly secure enclaves, or operating environments, for individual processes.

61. Due to expense, not all operating systems that followed implemented the entire combined hardware and software controls developed for MULTICS; however, the ideas of process separation and controlled communication between such processes continued to be crucial to systems security (J. Saltzer and M. Schroeder, The Protection of Information in Computer Systems, Proceedings of the ACM Symposium on Operating Systems Principles (SOSP), Oct. 11, 1974)—Ex-1032.

62. Advances in hardware made it possible for modern systems, including mobile systems, to implement more secure systems. For example, in the early 2000s, the Symbian mobile operating system (B. Li, et al., "Symbian OS platform security model", LOGIN: Magazine Vol. 35, No. 4, Aug. 2010—Ex-1033) separated processes into three "Trust Tiers": The Trusted Computing Base (TCB), The Trusted Computing Environment (TCE), and Applications, as illustrated below:



Ex-1045, 24 (Fig. 1)[1].

63. The first two layers/tiers shown were primarily dedicated to components of the operating system with relatively higher operating privileged, followed by the applications tier with applications having relatively lower privileges.

64. The first two tiers included the Trusted Computing Base and the Trusted Computing Environment. As shown, the Trusted Computing Environment includes, among other services, a Messaging service.

65. The Application Tier could be further subdivided into Signed and Unsigned applications. Signed applications could interact with the TCE and have

[1] All annotations added unless otherwise indicated.

this layer execute higher privileged requests, whereas Unsigned applications could not access that same functionality.

66. Through these mechanisms and others like them on more recent mobile platforms, it has long been possible for processes to operate in secure enclaves on mobile platforms, and with platform-supported encrypted communications across networks.

VI. THE '192 PATENT

A. Overview of the '192 Patent

67. The '192 Patent is directed to "[a] message link server that maintains secure message links with device link agents on each of a plurality of wireless end-user devices." Ex-1001 ('192 Patent), Abstract. It discloses that the "message link server" uses a "buffer" to store messages (like text and multimedia messages) from network elements and delivers the messages to end-user devices when some triggering event occurs. *Id.*

68. Figure 16 (reproduced below) provides an overview of a network including a server side (referenced as service controller 122), and a device side (referenced as device 100) that are in communication with each other through service control Link 1653:

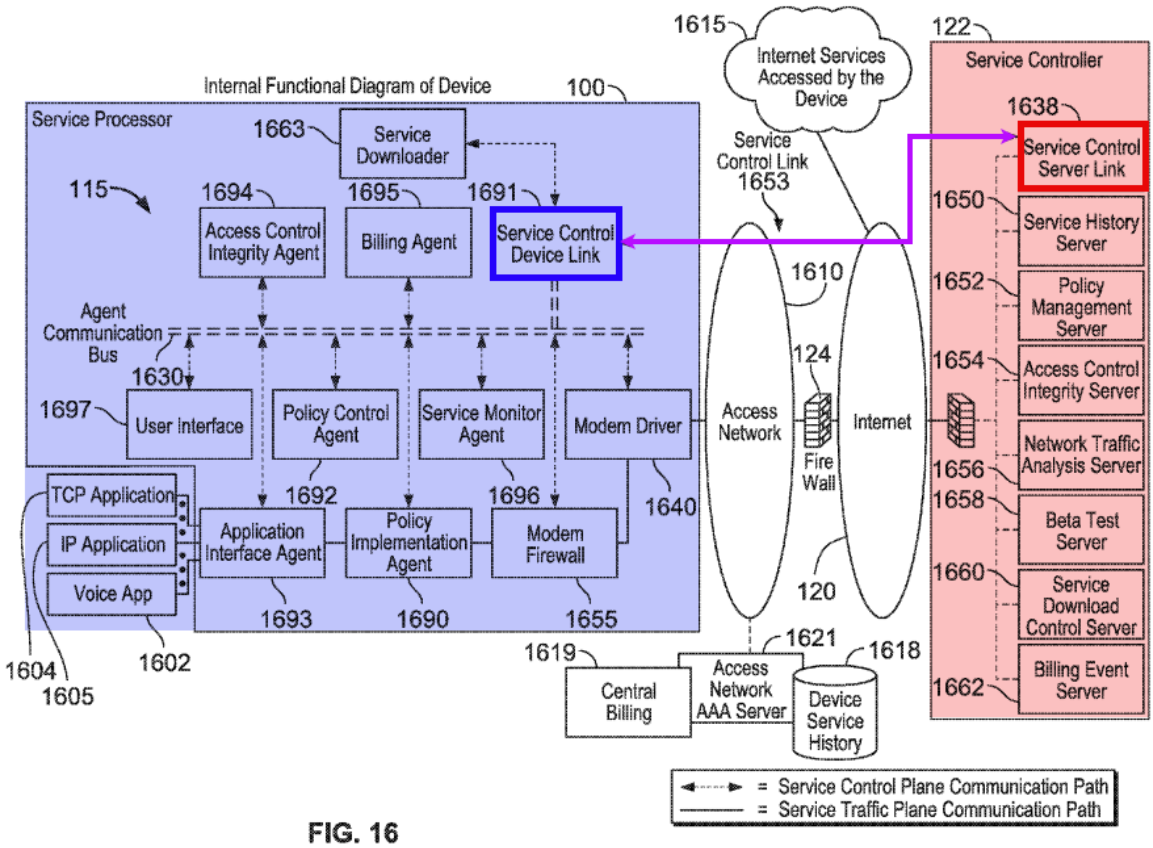


FIG. 16

Id., Fig. 16

69. Service Controller 122 includes multiple servers (e.g., 1650, 1652, 1654, 1656, 1662) that are communicatively coupled to the service control server link 1638. *Id.*, 68:19-20. On the device side, device 100 includes service control device link 1691, which is communicatively coupled to multiple device agents (e.g., 1697, 1695). *Id.*, 37:34-46; 41:45-53; 45:26-30.

70. To transmit messages to end-user devices, a secure link is established using protocols (or "stacks") between the "message link server" (red) and a "device link agent" (blue) on the end-user's device. *See, e.g.*, *Id.*, Abstract 89:11-90.4. Fig.

24 (reproduced below) "is a functional diagram illustrating the service control device link 1691 of [device 100] and the service control service link 1638 of the service controller 122." *Id.*, 87:49-52.

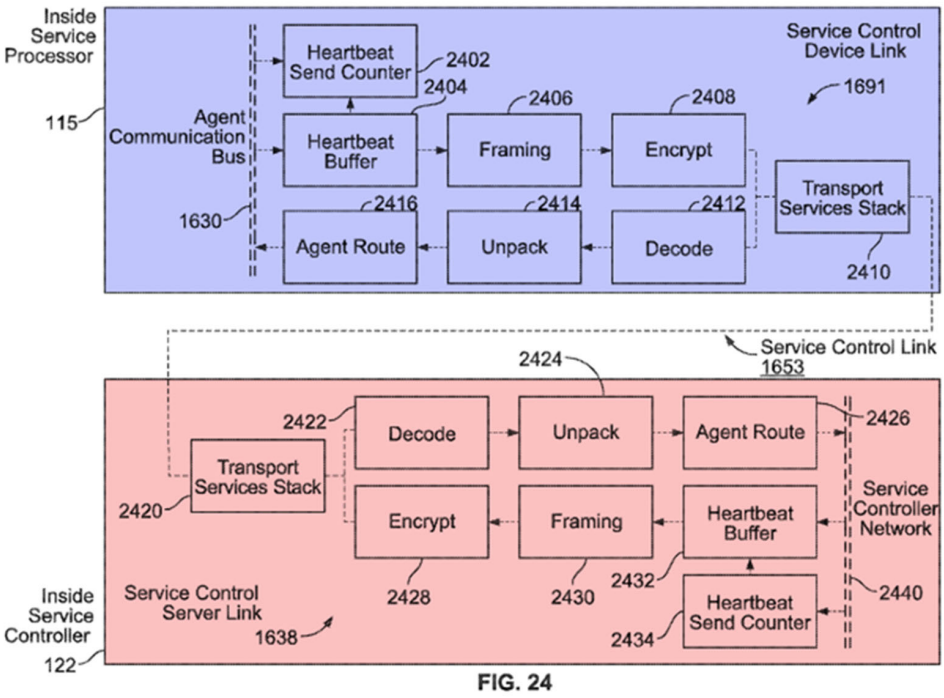


FIG. 24

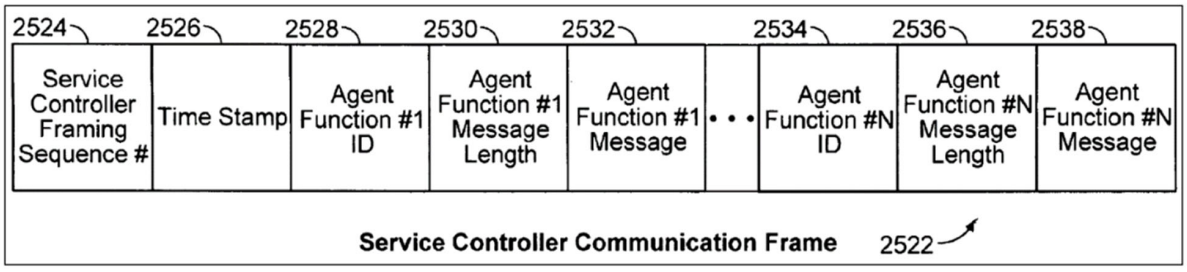
Id., Fig. 24

71. As shown, the service control server link 1638 includes a transport services stack 2420 that provides "basic transport services" and can be "implemented with standard secure or open Internet networking protocols, such as TLS or TCP." *Id.*, 89:29-36.

72. "Network elements send messages to" this server, with the messages being "targeted for delivery of message content to specific software components on respective ones of the wireless end-user devices." *Id.*, Abstract.

73. The messages are buffered "until one of several triggers occur, such that messages that do not have time-critical messaging needs may be buffered until the occurrence of a trigger other than the receipt of that message by the server." *Id.* The service control server link element 1638 can communicate one or more messages to the service control device link 1691. *Id.*, 91:51-92:19. A message received from server link element 1638 is associated with an identifier (e.g., "an agent first function ID 2528") and content (e.g., "an agent first function message 2532") included on the communication. *Id.*

74. The service control device link 1691 routes the content of the message "to the appropriate agent" on device 100 using the identifier. *Id.*, 90:52-91:23.



Id., Fig. 25 (excerpted).

B. Prosecution History of the '192 Patent

75. The application for the '192 Patent was filed on July 15, 2016, and claims priority to multiple non-provisional and provisional applications, with the

earliest of such applications being filed on January 28, 2009. Ex-1002 ('192 Patent), 730-33.

76. During prosecution, the Examiner initially rejected the then-pending claims on §112 indefiniteness and §103 obviousness grounds. *Id.*, 343-55. With respect to the obviousness rejections, the Examiner rejected the then-pending independent claims and certain dependent claims (claims 2, 5, 7-8, 10, 12-14, 16) in view of Heinonen (U.S. Patent No. 7,151,764) and Zhou (U.S. Patent Pub. No. 2006/0182137). *Id.*, 348-51; *see generally id.*, 343-55. The Examiner rejected the remaining dependent claims in view of Heinonen and Zhou, and further in view of the following references: Chattopadhyay et al. (US Patent No. 7,191,248), Pineau (US Publication No. 2003/0184793), Takeshima et al. (US Publication No. 2004/0054779), and Dilman et al. (US Publication No. 2002/0138599). *Id.*, 351-55.

77. In response, Applicant made minor amendments to the then-pending independent claims (*see id.*, 74-78), and argued that the proposed combination of Heinonen and Zhou failed to render obvious multiple limitations of the independent claims, which correlate to independent claim features of 1[pre], 1[a], 1[b], 1[c1], and 1[d1]-1[d3]. *Id.*, 81-86. As to the latter, Applicant advanced the following argument:

> *First, with respect to Heinonen, it is critical to note that Heinonen is directed to solving a problem with short-range wireless communications (such as Bluetooth) where a mobile device may not be in range of an access point*

long enough to establish a session, but the access point wants to quickly supply some useful information to passing mobile devices without establishing a session. An access point sends out Bluetooth inquiry packets indiscriminately to any mobile Bluetooth devices in range. (Heinonen, col. 12, 11. 1-3.) Any scanning device can choose to respond with an inquiry response packet that includes "class-of-device" information. (Id, 11. 13-27.) The access point can then respond with a paging packet, and according to Heinonen's technique, the paging packet includes a special "class-of-device" that notifies the responding device that a following paging packet with contain an Access Point Service Indicator ***(APSI)*** *message. (Id, 11. 53-58.) The access point performs a cache search for stored content appropriate for the mobile device's class-of-device type, and builds the special* ***APSI*** *message from whatever the cache hit returns. (Id, col. 14, 11. 40-53.) In the event that there is a cache miss, the access point enlists the help of a content server in order to locate appropriate content for the particular class-of-device type. (Id, 11. 53-66.) The access point then builds the APSI message and delivers it to the mobile device. (Id, col. 15, 11. 1-9.) The user device bypasses the protocol stack of the user device (since no session has been established with the access point) to deliver the APSI content directly to the GUI display function of the device. (Id., 11. 9-25.)*

Id., 81-82.

78. In view of this assertion, Applicant ***first*** argued that "Heinonen discloses no message link server because Heinonen's "access point generates its own APSI messages, when it chooses, either from cached content, or from a content server content, upon a cache miss, but has no mechanism to deliver a message from a network element to an identified software component on an identified device." *Id.*, 81-84.

79. ***Second***, Applicant argued that Heinonen discloses "no 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" because "during the time when the simple APSI message is communicated to a device, the access point and the device maintain no link at all (*see, e.g.*, col. 10, 11. 32-36, connection established at some point after **APSI** message is sent); nothing is secure as the content is in a paging packet; and nothing is through the Internet network, as these are directly communicated link-layer packets that bypass all 'middleware" protocol layers.'" *Id.*

80. ***Third***, Applicant argued that "Heinonen discloses no wireless end-user devices comprising multiple software components authorized to receive messages via the device link agent on that device" because, per Applicant, "[t]here is no mechanism to authorize other software components to receive messages, or direct messages to such components." *Id.*

81. ***Fourth***, Applicant argued that "Heinonen discloses no interface for network elements using the access point as a message link server" because Heinonen's "access point merely communicates with passing mobile devices opportunistically, nothing in Heinonen would lead one to expect that network elements would ever queue specific messages with an access point on the off-chance that a desired device passes within range; the content server merely supplies access-point requested content to an access point, it does not direct messages to the access point for delivery to respective software components on respective wireless end-user devices, as identified in the messages." *Id.*

82. ***Fifth***, Applicant argued that "Heinonen discloses no message buffering system" because "any APSI message delivered to a mobile device is built locally on-the-fly based on the cache content search and delivered immediately. The whole point of Heinonen it to be able to deliver a small amount of useful display content, quickly, to a passing device that is not connected to the access point." *Id.*

83. Applicant then argued that Zhou did not overcome the deficiencies in Heinonen. Ex-1002, 83. Specifically, Applicant argued that

> *As neither reference has a message link server, the combination would not teach or suggest to one of ordinary skill such a server. One reference [(Heinonen)] maintains no links-secure, for messaging, or through the Internet; the other reference [(Zhou)] is used within a single system*

and not on the Internet, and resorts to individual kernel-maintained connections between individual process pairs-thus the combination would not teach or suggest to one of ordinary skill the claimed transport services stack. One reference has no interface to receive messages from a plurality of network elements for delivery of respective message content for processing to respective ones of the software components identified in the messages; the other reference sets up many dedicated message channels between different pairs of programs running on the same system, but has no interface shared by multiple elements-thus the combination would not teach or suggest to one of ordinary skill the claimed interface to a network. One reference [(Heinonen)] teaches no buffering and a need for immediate delivery of a locally-generated message; the other reference [(Zhou)] teaches individual buffers for individual pairs of communicating programs, but no buffer shared for multiple sources that may be directing messages to a same remote device, where a message delivery trigger will cause the delivery of messages to multiple "consumers" on the same remote device.

Ex-1002, 83-84.

84. Applicant further argued that neither reference teaches the claim element reciting "as a trigger, the occurrence of an asynchronous event with time-critical messaging needs." Ex-1002, 84.

85. With respect to dependent claims, Applicant relied upon the arguments with respect to the independent claims and offered additional arguments for certain dependent claims. *See* Ex-1002, 84-86.

86. In the next Action, the Examiner allowed the application upon conducting an interview with the Applicant. The allowance included an examiner's amendment that amended the independent claim in the manner shown below (and made similar amendment to the other independent claim as well as conforming edits to certain dependent claims):

> **2. (Currently Amended)** A message link server comprising:
>
> 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, each of the wireless end-user devices comprising multiple software components authorized to receive **and process data from secure message link** messages **received via [[the]] a** device link agent on that device;
>
> an interface to a network to receive **network element** messages from a plurality of network elements, 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** ~~**for processing to respective ones of the software components identified in the messages**~~; and

a message buffer system **including a memory and logic, [[to]]**

the memory to buffer **content from** the received **network element messages for which delivery is requested** ~~**messages identified for delivery**~~ to a given one of the wireless end-user devices ~~**for which the respective secure message link is maintained**~~,

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 **[[the]] an** occurrence of an asynchronous event with time-critical messaging needs, and

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

Ex-1002, 13-19.

87. In the statement of reasons for allowance, the Examiner noted as follows: "The claims are allowable over the prior art of record in view of the claim

amendments above, for the reasons put forth in the Remarks (pages 9-11) of 02-01-2017" (which correspond to Ex-1002, 81-84) and "after further search and consideration, the prior arts of record either taken alone or in combination neither anticipates nor renders obvious the claimed subject matter of the instant application that is taken as a whole including the particular features incorporated in each independent claims." *Id.*, 18.

88. The application subsequently issued as the '192 Patent. *Id.*, 1.

C. Interpretations of the '192 Patent Claims at Issue

89. I have been informed by Counsel and understand that the best indicator of claim meaning is its usage in the context of the patent specification as understood by one of ordinary skill. I further understand that the words of the claims should be given their plain meaning unless that meaning is inconsistent with the patent specification or the patent's history of examination before the Patent Office.

90. Counsel has also informed me, and I understand that, the words of the claims should be interpreted as they would have been interpreted by one of ordinary skill as of the Critical Date (January 28, 2009). I have been informed by Counsel that I should use the Critical Date as the point in time for claim interpretation purposes. If the '192 Patent specification or file history informs a claim term's ordinary meaning, I address those teachings in the analysis below.

VII. THE ’320 PATENT

A. Overview of the ’320 Patent

91. The ’320 Patent concerns a “networked system” that uses a “buffer” to store messages (like text and multimedia messages) from network elements and delivers the messages to end-user devices when some triggering event occurs. Ex-1001 (’320 Patent), Abstract.

92. To transmit messages to end-user devices, a secure link is established using protocols (or “stacks”) between the “networked system” (red) and a “device link agent” (blue) on the end-user’s device. *See, e.g.*, *id.*, Abstract, 89:11-90:4.

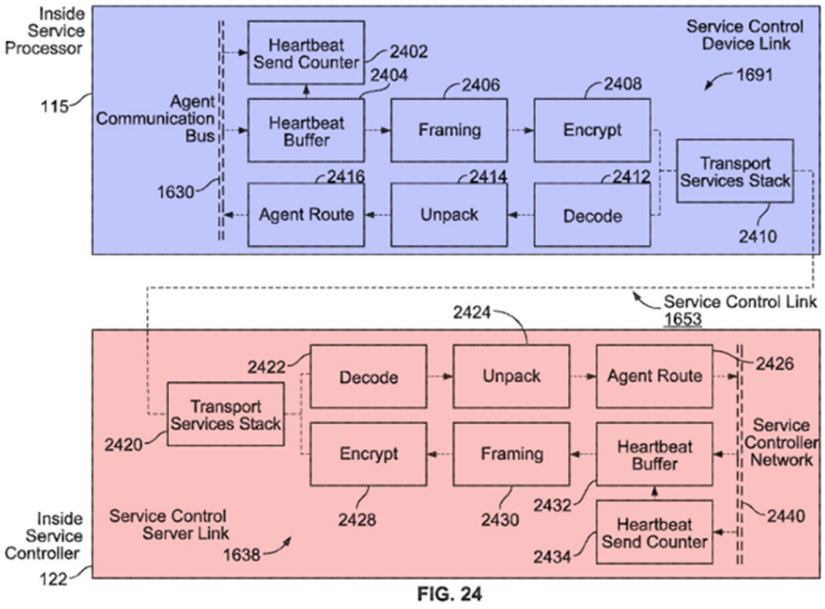


FIG. 24

Id., Fig. 24

B. Prosecution History of the ’320 Patent

93. The application for the ’320 Patent was filed on April 28, 2017, and claims priority to multiple non-provisional and provisional applications, with the earliest of such applications being filed on January 28, 2009. Ex-1001 (’320 Patent).

94. The original application included a single claim that was effectively directed to storing credentials for activating a communication device on a network. Ex-1002 (’320 Patent), 974. The Applicant later amended the claims, however, to cancel claim 1 and present 19 brand-new, much more specific claims. *Id.*, 474-78. The Examiner does not appear to have issued any rejection against the newly presented claims. *See generally* Ex-1002 (’320 Patent).

95. Instead, in the statement of reasons for allowance dated April 4, 2019, the Examiner simply noted that the claims were “allowed as all the cited prior art, either alone or in combination, [did] not disclose Applicant’s inventive claim language.” *Id.*, 13.

96. The application subsequently issued as the ’320 Patent. *Id.*, 2.

C. Interpretations of the ’320 Patent Claims at Issue

97. I have been informed by Counsel and understand that the best indicator of claim meaning is its usage in the context of the patent specification as understood by one of ordinary skill. I further understand that the words of the claims should be

given their plain meaning unless that meaning is inconsistent with the patent specification or the patent's history of examination before the Patent Office.

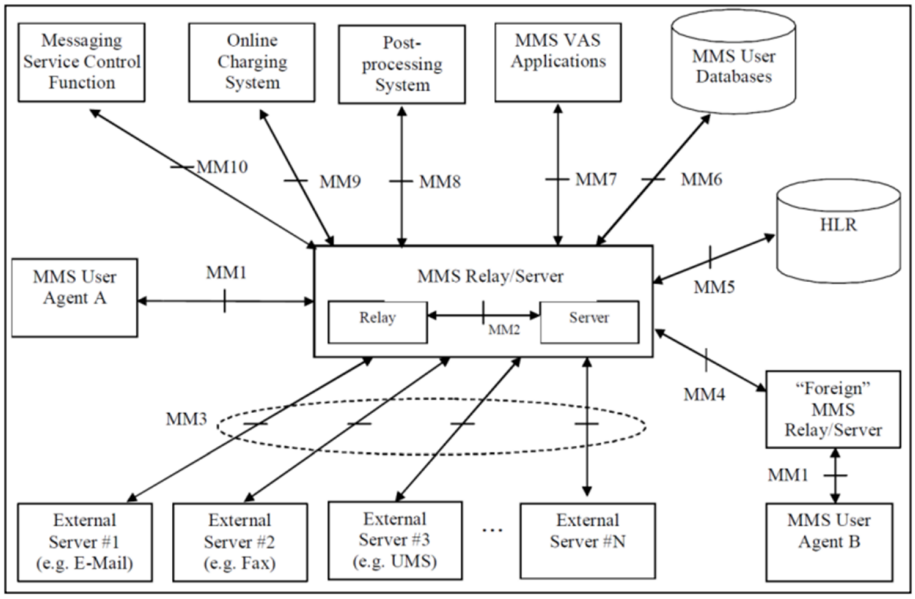
98. Counsel has also informed me, and I understand that, the words of the claims should be interpreted as they would have been interpreted by one of ordinary skill as of the Critical Date (January 28, 2009). I have been informed by Counsel that I should use the Critical Date as the point in time for claim interpretation purposes. If the '320 Patent specification or file history informs a claim term's ordinary meaning, I address those teachings in the analysis below.

VIII. OVERVIEW OF THE PRIOR ART

99. In my opinion, the prior-art references below are each analogous to the '192 and '320 Patents, and each other, because they are each directed to the same field of endeavor (e.g., computer systems and/or networks) or are reasonably pertinent to the problems faced by the inventor (e.g., computer messaging across a network and/or network/device security).

A. TS-23.140 (Ex-1004)

100. 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 describes "non-realtime Multimedia Messaging Service, MMS," with an example networked environment where MMS is implemented being shown below:



Ex-1004, Fig. 3.

101. In this environment, "MMS Relay/Server is responsible for storage and notification, reports, and general handling of messages" between different entities, including among others, an MMS User Agent, a Foreign MMS Relay/Server, and MMS VAS (Value Added Services) Applications. *Id.*, §5.2, p. 21. For example, MMS Relay/Server communicates with MMS User Agent through communication link MM1, and with MMS VAS Applications through communication link MM7. *Id.*, pp. 23-24.

102. "MMS User Agent [is an] application residing on a [user equipment (UE)] that performs MMS-specific operations on a user's behalf." *Id.*, p. 14. Communication link "MM1 is used to submit Multimedia Messages [(MM)s] from MMS User Agent to MMS Relay/Server, to let the MMS User Agent pull MMs from

the MMS Relay/Server, let the MMS Relay/Server push information about MMs to the MMS User Agent as part of an MM notification, and to exchange delivery reports between MMS Relay/Server and MMS User Agents." *Id*., §6.3, p. 24.

103. MMS VAS Applications are "[a]pplications providing Value Added Services (e.g. news service or weather forecasts) to MMS users." *Id*., p. 14. Communication link "MM7 is used to transfer MMs from MMS Relay/Server to MMS VAS applications and to transfer MMs from MMS VAS applications to MMS Relay/Server." *Id*., §6.9, p. 25. The received MMs are then "targeted to the recipient" MMS User Agent by the MMS Relay/Server and through MM1. *Id*., §7.1.13, p. 41.

104. "Apart from using MMS as a service for users to exchange messages, MMS may also be used to transport data specific to applications." *Id*., §7.1.18, p. 54-55. In particular, MMS can be "used to transport data specific to applications between two MMS User Agents or an MMS User Agent and an MMS VAS Application (or vice versa)." *Id.* at 55.

105. An application-specific message includes an "application identifier of the destination application" residing on the recipient user device. *Id*., p. 55. "Upon reception of an abstract message containing a destination application identifier[,] the receiving MMS User Agent…shall first check if the destination application resides on [the UE device]." *Id.*, §7.1.18.2.2, p. 56.

106. "If the destination application resides on a receiving MMS User Agent, the MMS User Agent shall immediately route the received MMS information on to the destination application that is referred to from the destination application identifier." *Id*.

107. TS-23.140 contemplates using security protocols such as Transport Layer Security (TLS) for its communication links. *See, e.g.*, *id.*, §7.1.13.1 ("For additional security, HTTP may be carried over a TLS [0066] session to the MM7 interface."). For example, TS-23.140 notes that "[d]etails for implementation of the MM1 transfer protocol are described in Annex B." *Id.*, §6.3, pp. 24-25. Annex B refers to Ex-1011, which in turn refers to Ex-1010 for implementation details of MM1. *Id.*, p. 162. Ex-1010 explains that "[t]he TLS [WP-TLS] transport layer security protocol provides for secure data transmission between the MMS Client and the MMS Proxy-Relay in architectural configurations that employ HTTP based protocol stacks for MMS_M implementation." Ex-1010, p. 22.

B. Shen (Ex-1014)

108. Like TS-23.140, Shen's "invention relates to multimedia message service (MMS), particularly a multimedia messaging service system, a gateway and a user device" and "a method for receiving/sending multimedia message." Ex-1014, [0001].

109. Shen explains that, "[b]ecause MMS is a message store/forward system, MMS messages have to be stored and forwarded by the MMS Center owned by a telecom operator when enterprise users communicate with the backend system (MMS applications 112) with a multimedia messaging user device (hereinafter referred to as 'MMS user device') via the MMS channel." Ex-1014, [0003], Fig. 1 (showing MMS Gateway including the MMS Server/Relay facilitating communication between MMS applications and an MMS User Device).

110. "Therefore there is a potential security problem in … [such] systems since it is not an end-to-end solution for enterprise mobile applications." *Id.*, [0004]. Shen's invention contemplates multiple different embodiments to solve this problem, one of which is described with reference to Shen's Figure 5 below.

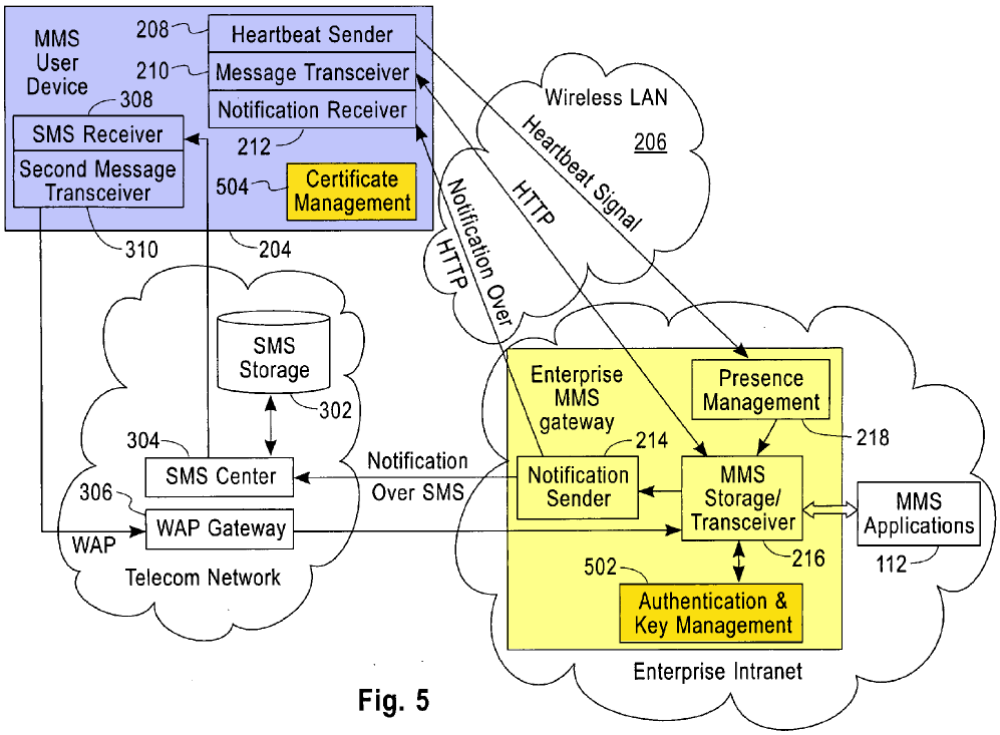


Fig. 5

Ex-1014, Fig. 5.

111. Specifically, in this embodiment, "an authentication and key management module 502 is added in the MMS gateway to distribute symmetric keys to users and a certificate management module 504 is added in the user device." *Id.*, [0054]. "Both the MMS gateway and the user device deploy the same symmetric cipher to encode and decode MMS messages to protect the privacy of these messages." *Id.* The encryption may be realized in the following manner:

- First, if "the MMS user device has no certificate of the MMS gateway, it … sends a CA (certificate) requirement to the [MMS] gateway. After the gateway confirms that the user has the access to send and receive

MMS according to the user device number, the [MMS] gateway's CA is sent to the user device…." *Id.*, [0056].

- Next, "the MMS user device sends a Master Key requirement to the gateway and the gateway creates a session ID and a master key, generally a random object, and sends them to the device via SMS or WAP response. The random object is encrypted by the gateway's private key." *Id.*, [0057].
- The "random object" is then "decoded by the user device and mixed with its phone number and session ID to generate a shared secret key…. At the same time, the gateway also generates a shared secret key with the same algorithm." *Id.*, [0058].
- And then, "[b]oth the MMS user device and the MMS gateway can secretly communicate with each other by encrypting MMS messages with the shared secret key." *Id.*, [0059].

112. Shen further contemplates that, in the above encryption process, both the MMS user device and the MMS gateway can update "the shared secret key by changing the session ID to prevent the threat of attacks stemming from key reuse." *Id.*, [0060].

113. In this manner, Shen's "invention … provides a MMS system supporting both telecom network and wireless LAN and having improved security,"

including "provi[sion] [of] an end-to-end security solution for MMS applications," which can favorably be implemented for use with "enterprise applications." *Id.*, [0017], [0021] (describing additional advantages stemming from Shen's invention).

114. Shen further discloses that the MMS User Device includes a heartbeat sender for sending heartbeat signals to be detected by a wireless LAN. *Id.*, [0018]. The heartbeat sender sends signals to the MMS gateway which records the transmitted information in the presence manager. *Id.*, [0034]. When "the gateway periodically receives the heartbeat signals, it means the corresponding user is located in the coverage of the wireless LAN, and the gateway may send a notification to the user when there is a message to be sent to that user." *Id.*

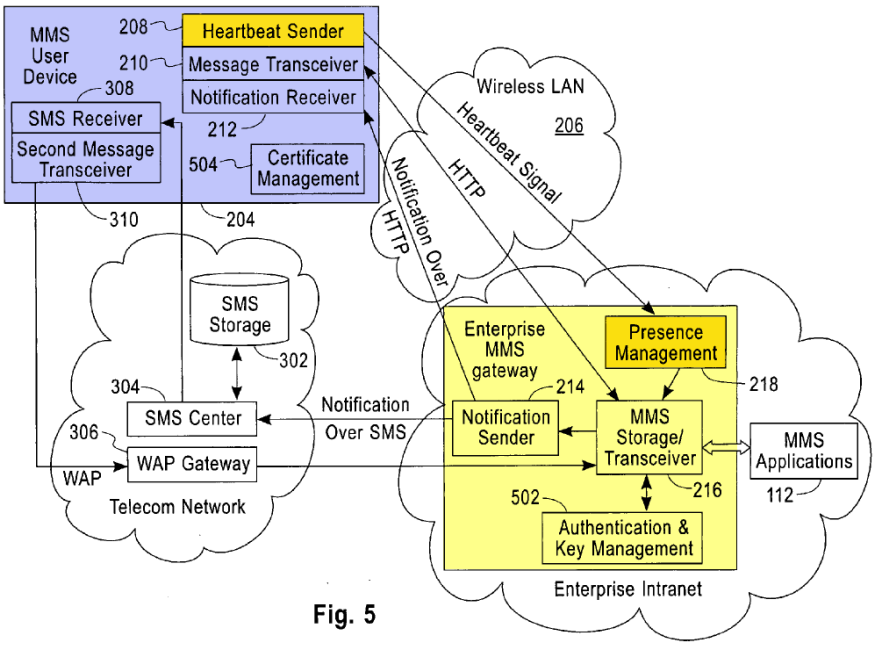


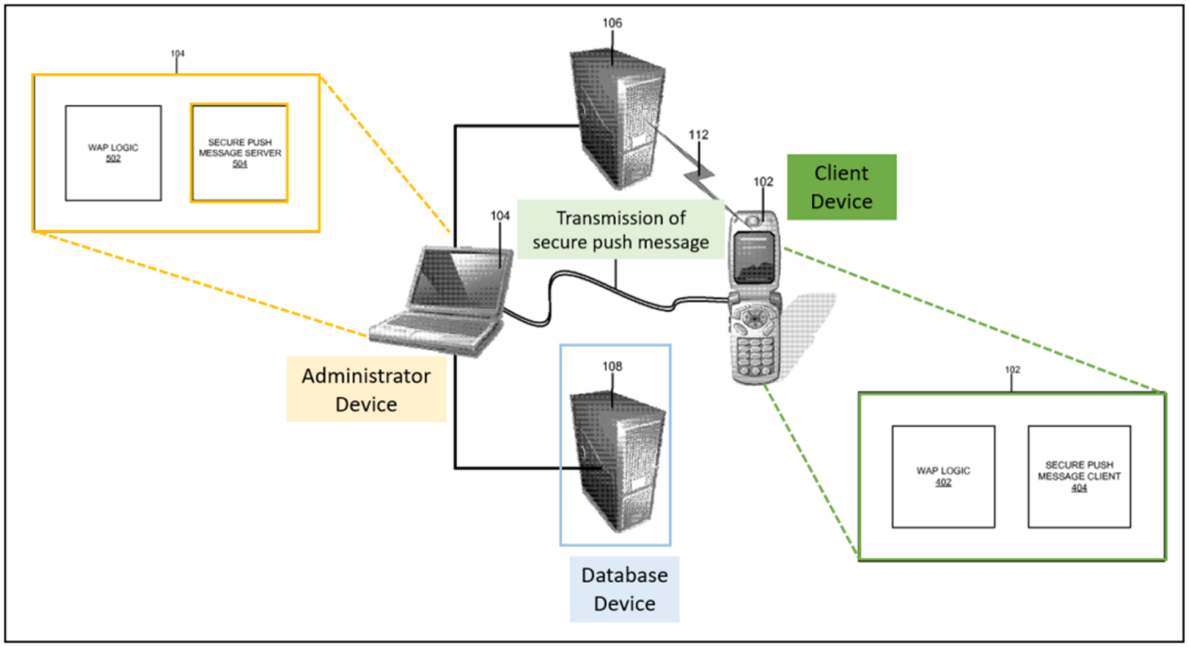
Fig. 5

Ex-1014, Fig 5.

C. Rakic (Ex-1015)

115. Rakic relates to secure push messages. Ex-1015, Title. Rakic discloses that, in the context of push communications, "a sender may send a secure push message to a client device," where the secure push message may bear an electronic signature that validates (e.g., authenticate) the secure push message at the client device. *Id.*, [0039].

116. Rakic discloses a networked environment in which an administrator device 104, which includes the secure push message server 504, communicates secure push messages over a network to a client device 102:



Ex-1015, Figs. 1, 4-5 (excerpted).

117. As shown, the environment additionally includes a database device 108 that stores information related to client devices (e.g., codes obtained from a client

device such as IMEI or a component in the client device (e.g., removable memory). *Id.*, [0047]-[0049].

118. The secure push message server 504 generates a key, an electronic signature, and a secure push message for a message from a particular user/administrator. *Id.* Specifically, the secure push message server generates the key based on the above-described code(s) (e.g., IMEI) that are obtained from client device 102 or a component thereof, and subsequently, generate an electronic signature based on the key and the message. *Id.*, [0066]-[0067]. In particular, the electronic signature is created by "generat[ing] a first key by combining an administrator code, a client device identifier that identifies a client device, and subscriber information that is associated with a service to which a user subscribes." *Id*., Abstract.

119. The sending device then "hash[es] the first key to generate a second key, and use the second key to sign a data block within the secure push message to produce an electronic signature." *Id*. The secure push message server 504 generates and sends the secure push message to client device 102, e.g., by appending the electronic signature to the message and/or other information (e.g., a timestamp). *Id.*, [0067], Fig. 8.

120. At the client device 102, a secure push message client 404 can validate the message and the sender by generating a key and then generating a signature using

this generated key and the data block of the received message. [0109]-[0112]. If the signature portion of the received message matches the signature portion of the received message, the secure push message is determined to be valid. *Id.*

121. In this manner, Rakic discloses that the secure push message server generates signatures for inclusion in push messages, and these signatures are then used by the (1) client device to "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." *Id.*, [0041].

D. Houghton (Ex-1007)

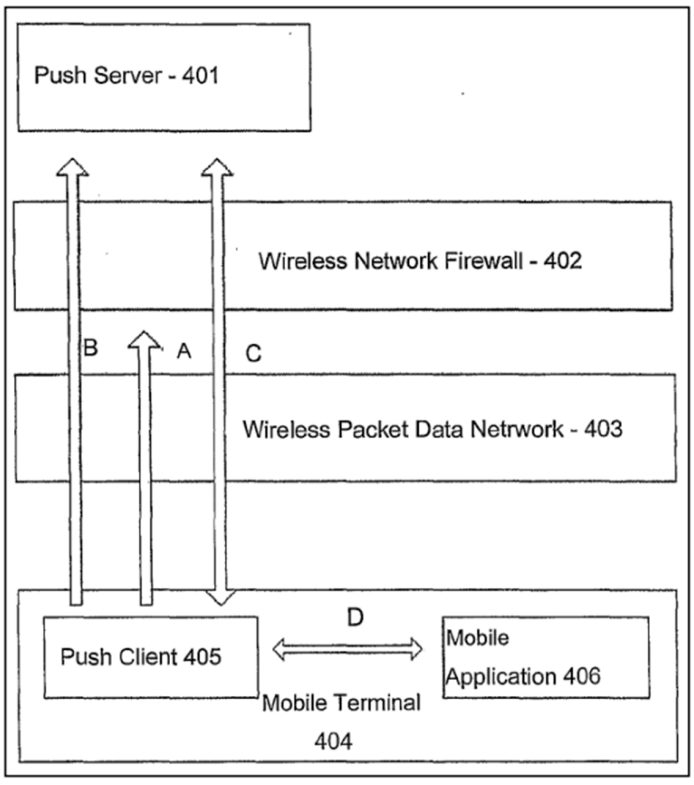
122. Houghton discloses "technique[s] for providing a push-type command of applications on a mobile terminal by a remote server computer." Ex-1007, 11.[2]

123. Houghton discloses a server that "push[es] messages to a" client on a "mobile terminal" in a wireless network. *Id.*, Abstract, 13. Push messages "may be triggered by any trigger event, local or remote, defined at the server," including an "alarm, notification, or measurement result received to the push server 401 from another device or system." *Id.*, 23. Upon triggering, messages are then transmitted using secure protocols, e.g., "HTTPS, IP-Sec, secure IP6 or a proprietary security protocol." *Id.*, 21.

[2] All references to page numbers in Houghton (Ex-1007) are to the page numbers printed on the document rather than to page numbers that will be stamped on the exhibit.

124. In some embodiments, Houghton seeks to push, from a server and to applications on a mobile terminal, messages/commands that "initiate control of one or more applications on an individual's mobile terminal." *Id*., 12-13. Based on the pushed commands, the server can deliver to the mobile applications executing on mobile terminals "real time information, interactivity and rich media content, regardless of which mobile operator is being used and where the mobile user happens to be at that moment." *Id*.

125. One example embodiment "for pushing mobile application launch commands to a mobile terminal" is shown in Figure 4, which is reproduced below:

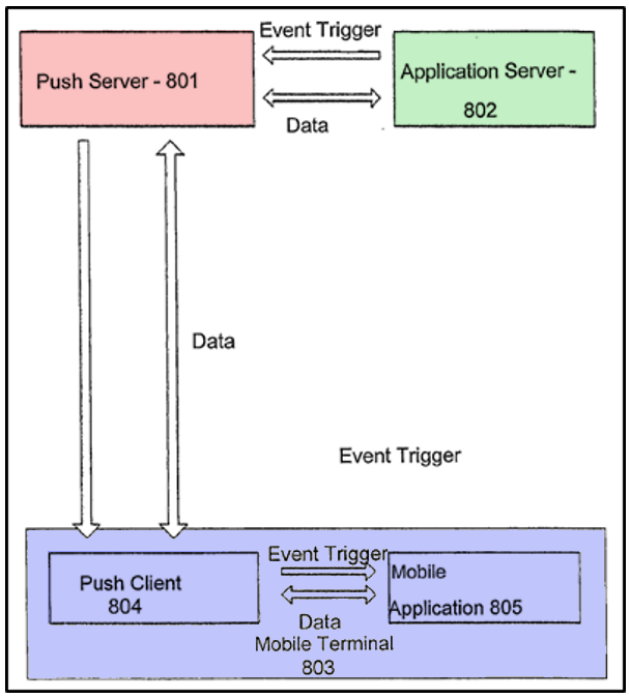


Ex-1007, Fig. 4

In this embodiment, after a "connection is established between the client 405 and the server 401," push client 405 pushes "a message to the" push server 401 or vice versa, as illustrated by arrow C. *Id*., 22.

126. The push message includes "application-specific" information "specifying which mobile application 406 from a plurality of such applications" is the target of the push message. *Id*., 23. When the mobile terminal receives this message, "appropriate automated actions and user interface media capabilities of the mobile terminal as specified in the command C from the server 401 are executed by the push client 405." *Id*. "The most common case is playing a sound, asking the user if the mobile application should be launched, and if affirmative[,] passing the command details (arrow D) to the specified mobile application." *Id*.

127. Figure 8 shows another embodiment that depicts the above-described communication between the mobile terminal and the push server 801, and further depicts "a data connection between application server 802 and mobile application 805," which is "established with the aid of the persistent managed, tested and configured data connection in Figure 4, arrow C between push server 401 /801 and push client 405/804." *Id*., 25.



Ex-1007, Figure 8

128. Houghton further discloses using a "connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS" to facilitate a secure communication link between the push server and mobile terminal. *Id.*, 20. Houghton also contemplates "exchanging security keys" between push client and push server. *Id*, 29.

E. Munson (Ex-1017)

129. Munson, like Houghton, discloses a method of "pushing contents to client devices." Ex-1017, Abstract. Munson, however, discloses "group pushes" where 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, and various message queuing policies. *Id.*, Abstract, [0037],

Figs. 1-5. Illustrated below is an example process according to Munson, showing buffering of received push messages:

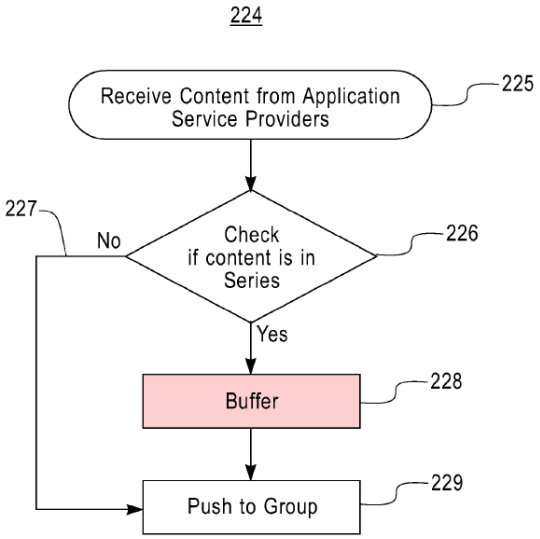


FIG. 4

Id., Fig. 4.

F. Ellison (Ex-1019)

130. Ellison discloses a "method and apparatus to protect a subset of a software environment." Ex-1019, Abstract. Ellison selectively isolates different software elements using a hierarchy of abstract rings and "nubs" to create an "isolated execution mode" (shown in red below) in which hardware/software access "is restricted" compared to a "normal execution mode" (outside of the secure-execution environment, green below) that "operates in a non-secure environment."

Id., 4:65-5:1, 6:1-26, 8:25-32, Figs. 1A-1C. Ellison controls access using computer-generated keys. *See, e.g.*, *id.*, 8:66-9:40, 9:47-62, Fig. 2.

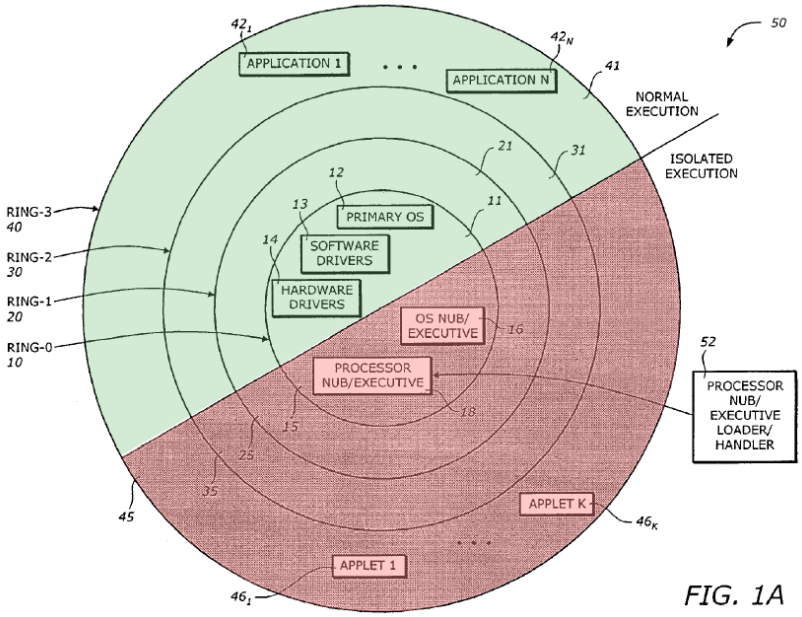


FIG. 1A

Ellison, Fig. 1A.

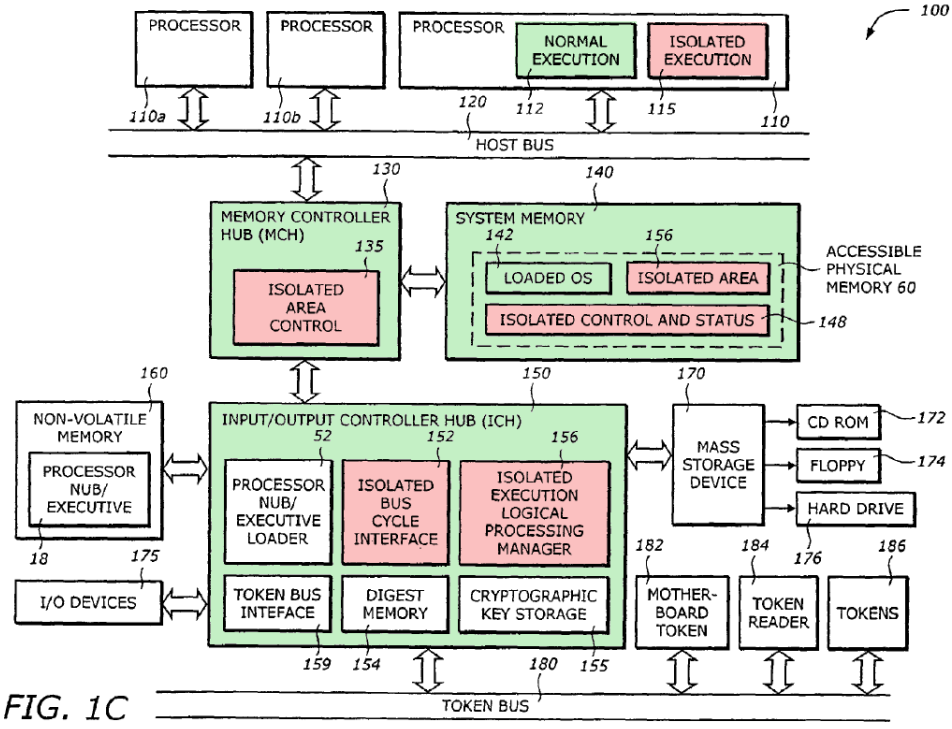


FIG. 1C

Ellison, Fig. 1C.

G. Adamczyk (Ex-1021)

131. Adamczyk discloses a system intended to provide a more-unified networking system to enable interactions between different applications over a network (much like MMS). *See* Ex-1021 (Adamczyk), Abstract, [0004] ("most applications today are 'atomic' in nature…with its own interface and other ancillary features like authentication and/or authorization. [Next-generation networks] may begin to change this paradigm" by enabling applications to "become increasingly more intertwined and composite, and hence more useful to the end user. Thus one may be able to invoke feature-rich multi-modal communication capabilities."), Figs. 5, 20.

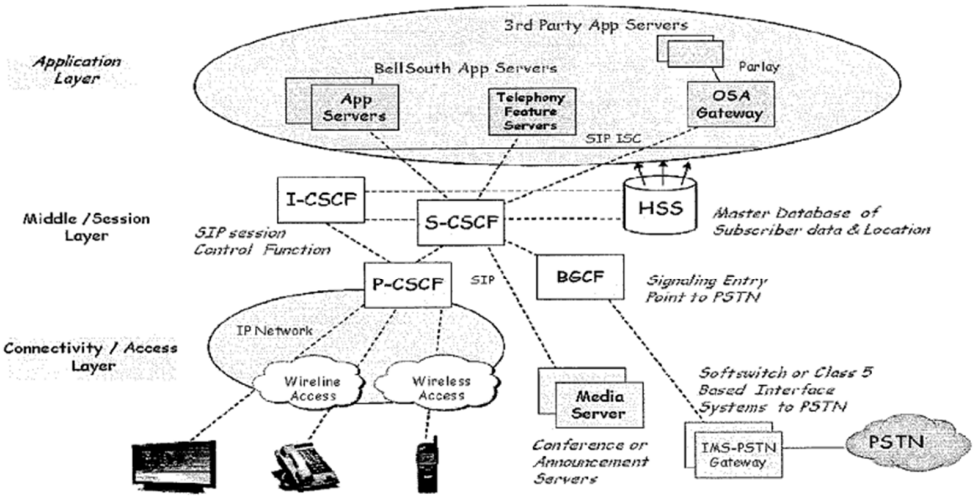


FIG. 5

Ex-1021, Fig. 5

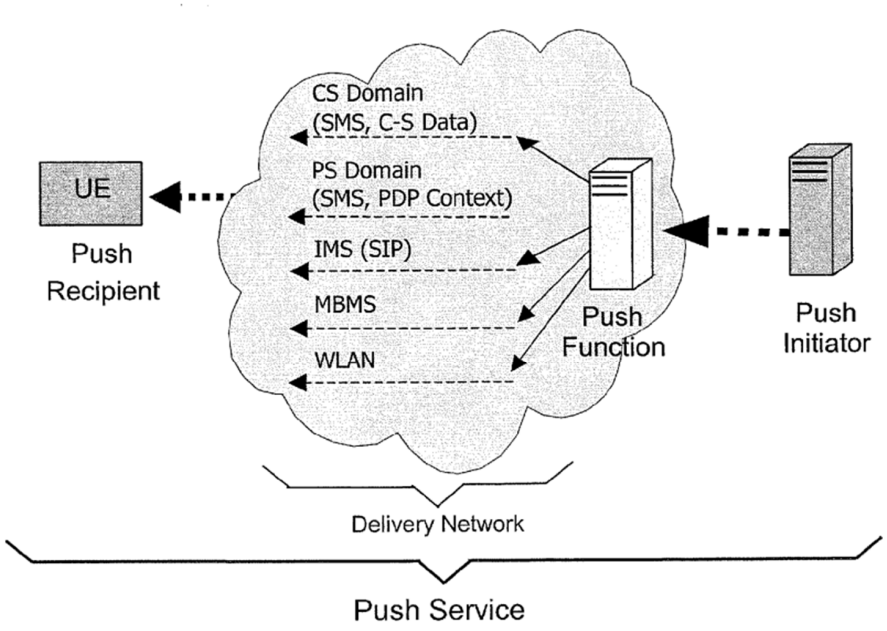


FIG. 20

Ex-1021, Fig. 20

132. As part of this system, Adamczyk envisions enabling applications to issue notifications across the network using a "notification server," which can be configured to send notifications "on demand, at a specific future time, and/or on a periodic schedule." Adamczyk, [0010], [0022]. Adamcyzk also envisions these features being implemented in standardized "3GPP network push service elements." *Id.*, [0017].

H. Herzog (Ex-1008)

133. Herzog describes a system and method for pushing and pulling messages between a server and a client through a gateway. Ex-1008 (Herzog), Abstract, Fig. 2.

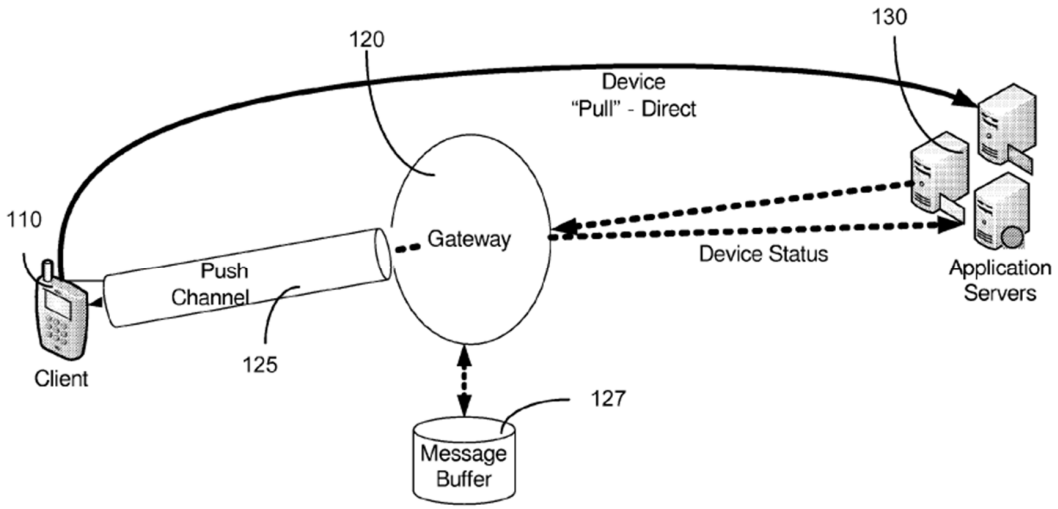


FIG. 2

Ex-1008, Fig. 2.

134. Herzog further describes that the UE device sends a periodic "keep-alive" message to the server or the gateway to make sure that the connection between the UE device and the server is continually maintained. *Id.*, [0036], [0041]-[0043]. So long as the message is sent on time, based on some timing parameter set at the server, the connection between the client and the server will stay open, but once the gateway does not receive a keep-alive message on time, the connection can be set to disconnected or otherwise terminated.

I. Gellens (Ex-1020)

135. Gellens describes a link between two entities, wherein the link between the two entities is maintained to allow "opportunistic" use of the link. Ex-1020 (Gellens), [0016], [0024]. Multiple applications on an entity, such as a user device,

can be capable of communicating over the link. *Id.*, [0025]. The link is configured such that there are a set of triggering criteria that may allow a particular application to communicate over the link. *Id.*, [0025]. Various priority levels are noted between the different applications, allowing some level of channel coordination. *Id.*, [0035]. When a high priority application initiates a traffic channel, other applications can be notified of this, and can also take use of the traffic channel, making an efficient traffic channel for multiple applications. *Id.* The traffic channel can be configured to notify other applications once a particular bandwidth has been reached, to allow for maximum advantage of piggy-backing low priority usage. *Id.*

J. Pazhyannur (Ex-1037)

136. Pazhyannur describes a communication system involving transmissions between devices and servers, including transmissions from applications. *See* Ex-1037 (Pazhyannur), Abstract.

137. As part of the system, Pazhyannur discloses that the system may store user profiles "for each user subscribed to the services of communication system 100. Each profile includes a list of user terminals registered by that user, for example, an identifier associated with each such device." *Id.*, [0040]. The profiles can also include "a list of communication devices that the user has permission to access, for example, to copy applications to or from or to transfer applications to or from such

devices" to provide access to a server and "retrieve a list of application ensembles created by any of the other devices registered by, or accessible by, that user." *Id.*

K. Fok (Ex-1038)

138. Fok describes techniques for "secure inter-application communication" in a "mobile operating environment." Ex-1038 (Fok), [0036]-[0037], [0081], [0106]-[0108], Figs. 4, 8-9. Fok uses a "round trip" concept where a "handshake" is established between a "primary application" and one or more "recipient" applications. *Id.*, [0047]-[0053], [0096]-[0102], [0106]-[0108], Figs. 7-10. Applications verify that another application is trusted during the handshake process using "a list of unique identifiers" associated with trusted applications installed on the mobile device. *Id.*, [0048]-[0050], [0096], [0101]-[0102], [0106], Figs. 8-9.

IX. ANALYSIS OF THE '192 PATENT

A. Ground 1A: Claims 1, 5, 6, 7, 9, 11-13, and 15 Are Rendered Obvious By TS-23.140[3]

1. Claims 1 and 15

*a. **1[pre]**: A message link server comprising / **15[pre]**: A method of operating a message link server, comprising:*

139. If the preamble is limiting, in my opinion, TS-23.140 discloses or renders obvious a message link server (e.g., the MMS Server/Relay) and a method for operating the same.

140. TS-23.140 discloses an MMS Relay/Server that facilitates delivery of messages between different entities over a network (as shown below):

[3] I note that the above descriptions of the references and the below provided combinations and analysis with respect to those combinations are incorporated into each mapping that includes citations to them. Additionally, all emphasis included in this declaration has been added unless otherwise indicated.

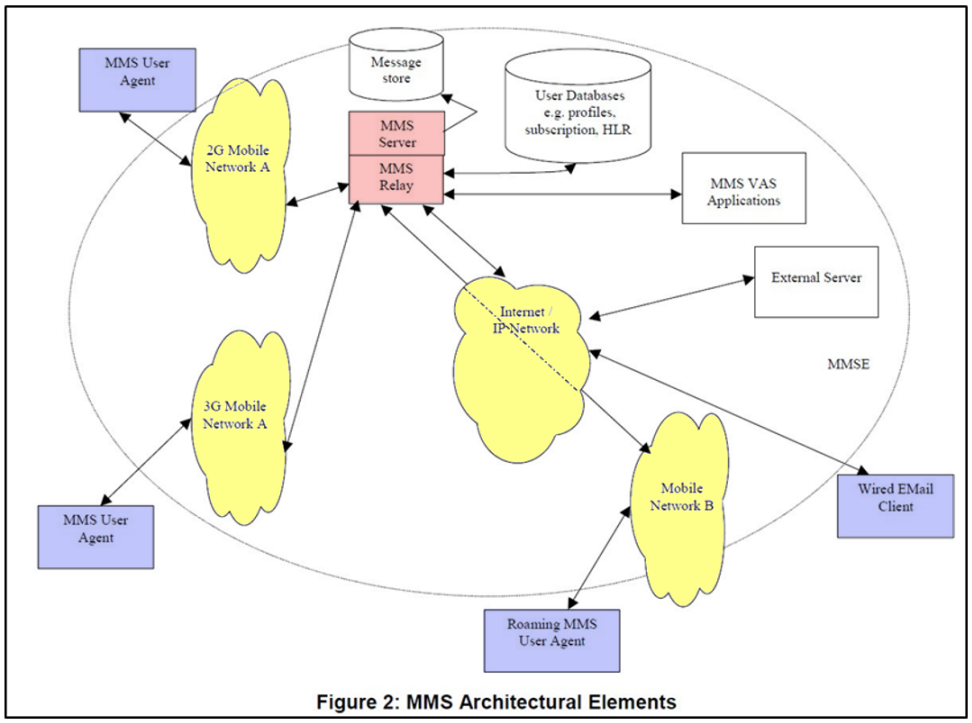


Figure 2: MMS Architectural Elements

Ex-1004, Fig. 2 (providing an MMS Relay/Server that facilitates delivery of messages between user devices and other elements in the network).

141. The MMS Relay/Server "is responsible for storage and handling of incoming and outgoing messages and for the transfer of messages between different messaging systems." *Id.*, 17 (noting that this server may be a single logical element or may be separated into MMS Relay and MMS Server elements; describing generation of charging data records (CDR) when retrieving or delivering multimedia messages (MM) and for Value Added Service Provider (VASP) operations).

142. As shown above, MMS communications are facilitated between MMS User Agents (i.e., between two user equipment (UEs)) or between MMS User Agents and MMS VAS Applications.

143. An MMS User Agent is an application residing on a UE, whereas an MMS VAS Application is an application that provides Value Added Services (e.g., news service or weather forecasts) to MMS Users and resides on a separate network entity, as shown in the above figure. *Id.*, 14.

144. MMS can also be "used to transport data specific to applications." *Id.*, 54. Such applications can be downloaded "to a mobile phone" and can register with an MMS User Agent of the UE (user equipment) or an MMS VAS Application. *Id.* Once registered, MMS can be used "to transport 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.

145. Additional details regarding the MMS Relay/Server and the method for operating it are provided below with reference to 1[a]/15[a]-1[d4]/15[d3] *infra*.

146. For the above reasons, TS-23.140 discloses or renders obvious 1[pre]/15[pre].

> *b.* ***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, /* ***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,*

147. In my opinion, TS-23.140 discloses or renders obvious a transport services stack (comprising the MM1 Transport Protocol and TCP/IP and transport

layer security (TLS)-based transport protocols) to maintain a respective secure message link (e.g., secure links) through an Internet network between the message link server (e.g., Message Relay/Server) and a respective device link agent (e.g., MMS User Agent) on each of a plurality of wireless end-user devices (e.g., wireless client devices).

148. In more detail, and as shown in Figure 2 (reproduced below), the MMS Relay/Server communicates via an Internet network with MMS User Agents that execute on each UE (wireless end-user device).

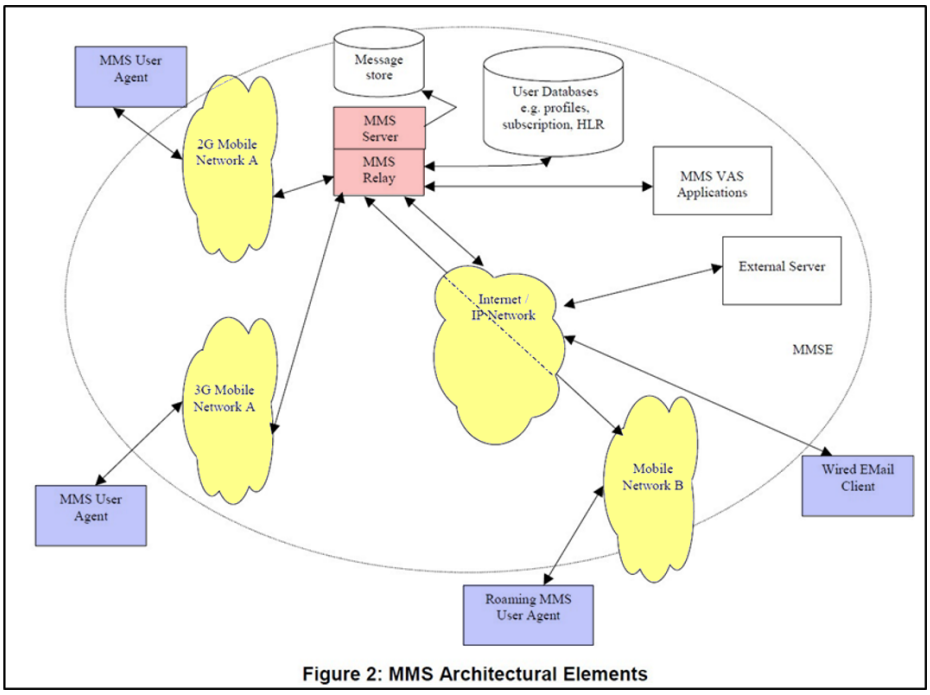


Figure 2: MMS Architectural Elements

Ex-1004, 17 ("Figure 2 shows that multimedia messaging may encompass many different network types … provided by the Internet protocol and its associated set of messaging protocols" that "enables messaging in 2G and 3G wireless networks to

be compatible with messaging systems found on the Internet."), 19-20 (describing MMS User Agent facilitating transmissions over a network).

149. Similarly, as disclosed in Ex-1010, Figure 3 depicts a similar MMS communication between an MMS client and the MMS Relay/Server over the Internet and a wireless network.

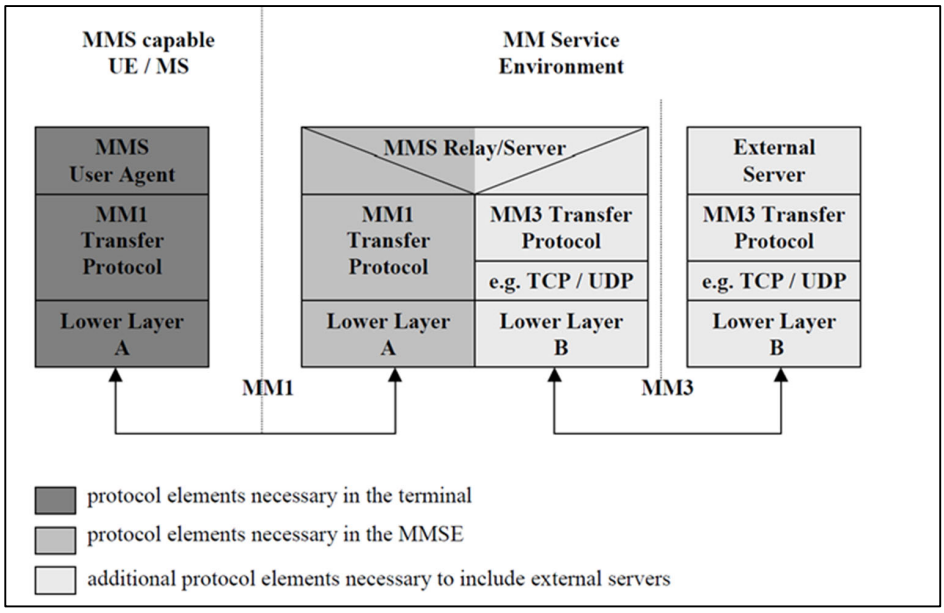
150. The MMS User Agent is a device link agent because it facilitates transmission and receipt of multimedia messages (MMs) between different devices (e.g., MMS User Agents executing on other UEs, MMS Relay/Server, MMS VAS Applications) that are linked together over a communication network (as shown in Figure 2 above). Ex-1004, 19-20, §5.1.1 (describing operations of MMS User Agent); *see also* Ex-1036, 2, Fig. 1 (showing multiple MMS User Agents in communication with MMS Relay/Servers).

151. The '192 patent does not specifically describe the structure of the transport services stack that facilitates the claimed communication link, and instead, functionally describes this claim element. For example, the '192 specification states that the "transport services stack" facilitates communications between the client and server devices and can be "implemented with standard secure or open Internet networking protocols, such as TLS or TCP." *See* Ex-1001 ('192 Patent), 89:24-41, 90:34-50.

152. A POSITA would have understood TLS to refer to Transport Layer Security, and TCP as referring to Transmission Control Protocol. *See* Ex-1027[4], Ex-1028.

153. Consistent with this disclosure, TS-23.140 also discloses or renders obvious a transport services stack in the form of a transfer protocol and associated functionality (that uses TCP and TLS) that maintains a secure message link through an Internet Network between the message link server (e.g., MMS Relay/Server) and wireless end-user devices (e.g., UEs using the same protocol). For example, TS-23.140 discloses that MMS shall "make use of the protocol framework outlined in figure 4" pursuant to which the MMS User Agent communicates with MMS Relay/Server. Ex-1004, §6.2, pp. 24-25.

[4] Ex-1027 refers to IETF RFC 793, Transmission Control Protocol (Sept. 1981) (available at https://www.ietf.org/rfc/rfc793.txt) and Ex-1028 refers to The TLS Protocol Version 1.0 (Jan. 1999) (available at https://datatracker.ietf.org/doc/html/rfc2246).



Ex-1004, §6.2, pp. 24-25, Fig. 4 (providing protocol framework (or transport stacks) to provide MMS; listing MM1 transfer protocol as part of MMS Relay/Server).

154. As shown, TS-23.140 discloses that the MM1 Transfer Protocol is implemented both at the MMS Relay/Server and the MMS UE. *See id.* A POSITA would have understood that the MM1 Transfer Protocol is part of a transport services stack that facilitates transmission/transport of network communications between the MMS Relay/Server and network elements (e.g., the MMS User Agent, MMS VAS Applications).

155. A POSITA would have understood that TS.23.140 discloses a transport services stack comprising the MM1 Transfer Protocol with TCP/IP and TLS protocols given its well-known description in the art as the MM1 Transfer "protocol ***stack***" that provides a "transport mechanism" using the HTTP/TCP/IP. Ex-1013, 11.

156. Indeed, consistent with this understanding, TS-23.140 contemplates "WAP/OMA implementation" for the "MM1 Transfer Protocol" and incorporates by reference Open Mobile Alliance specifications, such as those provided in Ex-1011. *See* Ex-1004, 13, 162. This incorporated document explains that "[s]ince MMS is an application-level protocol framework to enable messaging transactions, a device implementing OMA MMS must have either of WAP WSP stack or HTTP/TCP/IP stack." Ex-1011, 11.

157. Further still, the same suite of OMA specifications related to MMS explains that "TLS" or "transport layer security protocol provides for secure data transmission between the MMS Client and the MMS Proxy-Relay in architectural configurations that employ HTTP based protocol stacks for MMS_M implementation." Ex-1010, 22.

158. Given these disclosures and the well-known use of TLS to secure/encrypt network communications, a POSITA would have understood or found it obvious that the transport services stack in the MMS Relay/Server would use TLS for securing the communication link between the MMS User Agent and the MMS Relay/Server. This is further corroborated by Ex-1010, which contemplates "encrypt[ing] the communication channel between the client and the backend MMS server with some security protocols, e.g., WTLS (Wireless Transport Layer Security)." Ex-1014, [0017].

159. A POSITA would have further understood or found it obvious that in the MMS environment, multiple MMS User Agents (and their associated UEs) are in communication with the MMS Relay/Server, such that a separate secure TLS-based link is maintained between each MMS User Agent and the MMS Relay/Server. This is further corroborated by multiple references. As a starting point, a POSITA would have known that, in the relevant timeframe, TLS or SSL-based communications were expensive to form and thus used keep-alive and session-resumption methods to maintain already-formed/established connections (and avoid cost and time to re-establish such connections). Indeed, this is corroborated by Houghton, which discloses connections between push servers, akin to MMS Relay/Server, and push clients, akin to MMS User Agents, being "persistent managed, tested and configured" and use periodic messages to prevent "time expir[ing]" the connection. Ex-1007, 23, 17. For example, Mostafa discloses and shows (as reproduced below) communication links formed and maintained between MMS Relay/Server and multiple terminals (and their respective MMS User Agents):

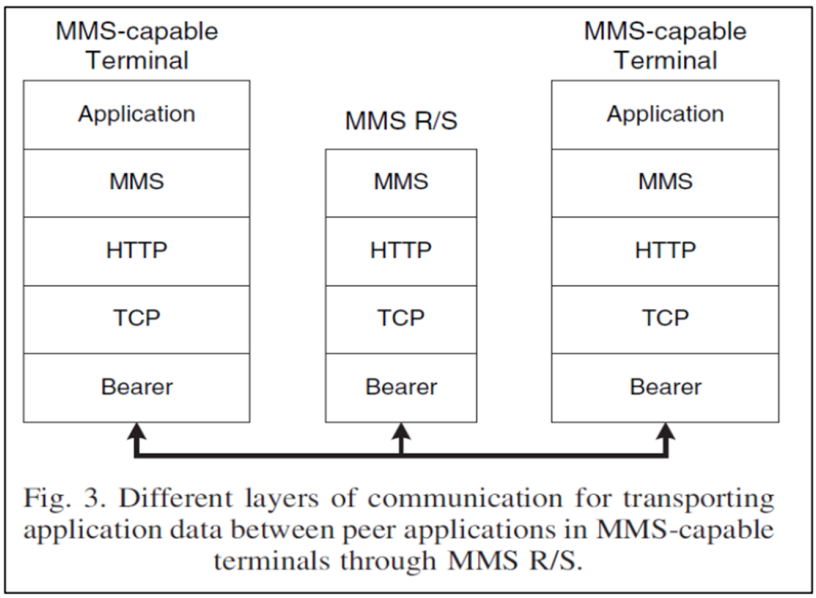


Fig. 3. Different layers of communication for transporting application data between peer applications in MMS-capable terminals through MMS R/S.

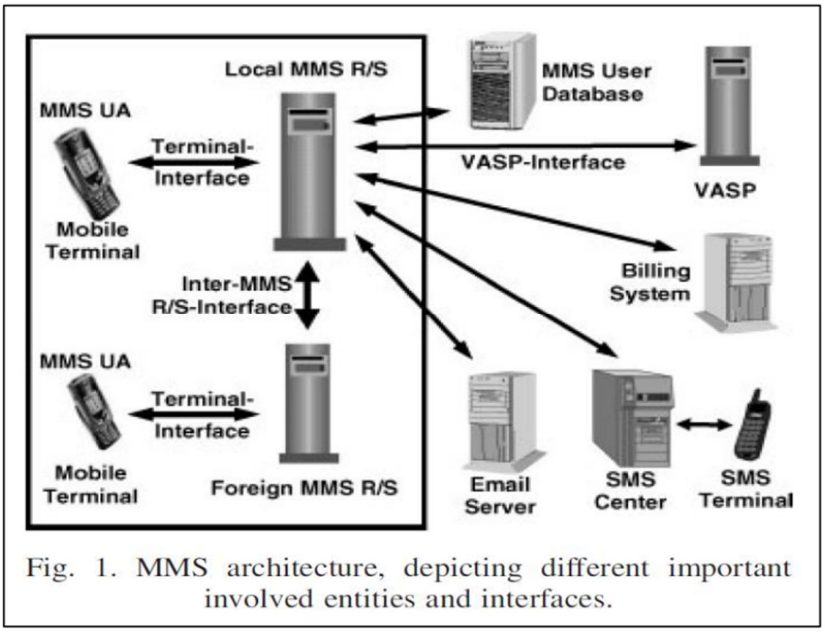


Fig. 1. MMS architecture, depicting different important involved entities and interfaces.

Ex-1036, Figs. 1-3, pp. 2-4. Munson further corroborates that such persistently maintained communication links are formed between multiple devices and the servers, as shown below.

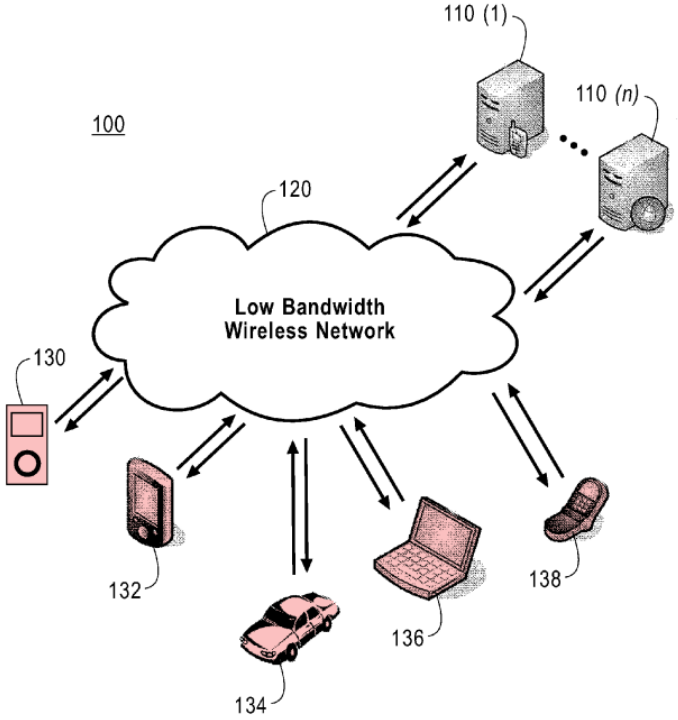


FIG. 1
Prior Art

Ex-1017, Fig. 1, [0007]-[0008] (showing plurality of data connections with multiple end-user devices).

160. For the above reasons, TS-23.140 discloses or renders obvious 1[a]/15[a].

c. ***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; /* ***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;*

161. TS-23.140 discloses or renders obvious that each of the wireless end-user devices (e.g., wireless client devices) include multiple software components (e.g., multiple applications) authorized to receive and process data (e.g., application-specific data intended for the application identified by the identifier and assuming user agent supports transport of application-specific data) from secure message link messages received via a device link agent on that device.

162. In more detail, TS-23.140 discloses that applications, such as those that are downloaded on a mobile phone, can use MMS to "transport data specific to ***applications***" and MMS communication can include "transport data specific to applications between two MMS User Agents or an MMS User Agent and an MMS VAS Application." Ex-1004, §7.1.18, pp. 54-55. Thus, TS-23.140 discloses that each UE (wireless end-user device) includes multiple software applications (software components). Ex-1036, 2-4 (showing and listing multiple applications in UEs that receive application data via MMS), Fig. 5 below.

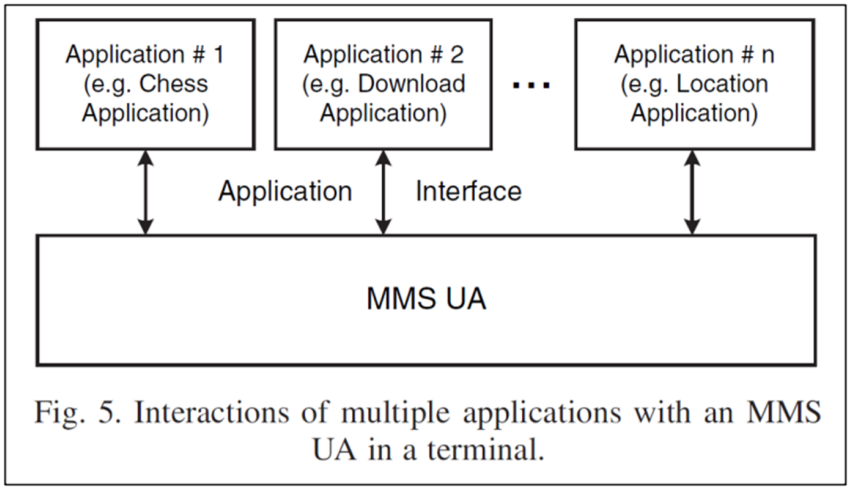


Fig. 5. Interactions of multiple applications with an MMS UA in a terminal.

163. A POSITA would have understood or found it obvious that each such application has to be authorized to receive and process messages via MMS. TS-23.140 discloses that "[a]pplications that want to transport data specific to applications other than MMS will initially ***need to register with the appropriate MMS User Agent or MMS VAS Application***." Ex-1004, §7.1.18, pp. 54-55. "During this registration process the application provisions an MMS User Agent or an MMS VAS Application with its application identification value and may negotiate with the MMS User Agent or MMS VAS Application the details (amount and format) of information to be exchanged between the two entities." *Id.*

164. Additionally, the MMS User Agent can indicate the "Terminal Capability" toward its recipient MMS Relay/Server by transmitting "a set of information describing the terminal's capabilities, including a binary flag indicating whether or not the terminal supports Application data." Ex-1004, §7.1.3.1, p. 30.

165. Subsequently, a message including application data for an intended application is only delivered to that application upon determining that the MMS User Agent has the capability of supporting application data and the terminal includes the intended application for the receipt of such application data. *Id.*, pp. 30, 55-56 ("If the MMS Relay/Server finds from the recipient MMS User Agent's capability indication … that the recipient MMS User Agent supports transport of application data, the MMS Relay/Server … shall pass on the destination application identifier, the 'reply-path' identifier (if present) and the additional application/implementation specific control information (if present) unaltered."; "If the destination application does not reside on the receiving MMS User Agent or on the USIM or MMS VAS Application, the MMS User Agent or MMS VAS Application shall discard the corresponding abstract message."). From these disclosures a POSITA would have understood that such applications are ***authorized*** to receive messages including application data.

166. A POSITA would have further understood that application data received by a particular destination application would be processed by that application to, e.g., facilitate display of that data or perform operations with respect to that data. Ex-1004, § 7.1.18.2.2, 56 ("rout[ing] the received MMS information on to the destination application" and contemplating "further handling and processing of the" received application data "by the destination application").

167. Further still, because the above-described communication is facilitated between MMS User Agents and/or an MMS User Agent and an MMS VAS Application, and such communications occur via the MMS Relay/Server (as described above in 1[a]/15[a]), a POSITA would have understood that the messages including the application data are received via the MMS User Agent (device link agent) on the particular terminal/device. Ex-1004, §7.1.18, pp. 54-55 ("Based on the negotiated details upon application registration process an application may trigger an MMS User Agent or an MMS VAS Application to submit certain abstract messages. Upon triggering an MMS User Agent or an MMS VAS Application to send an abstract message the MMS User Agent or MMS VAS Application may receive information from the application. ***The MMS User Agent or MMS VAS Application may insert this information in both the information elements and/or payload (if present) of the abstract message…. Abstract messages that are sent by an MMS User Agent or an MMS VAS Application on behalf of an originating application shall contain a destination application identifier***."); *id.*, 56 ("Upon reception of an abstract message containing a destination application identifier (it can either be the MM1_notification.REQ, MM1_retrieve.RES or MM7_deliver.REQ transactions), ***the receiving MMS User Agent or MMS VAS Application shall first check if the destination application resides on it***" and if so, the "*MMS User Agent shall immediately route the received MMS information on to*

the destination application that is referred to from the destination application identifier").

168. As I explained above in 1[a]/15[a] (and incorporate here), "TLS provides for secure data transmission between the MMS Client and the MMS Proxy-Relay in architectural configurations that employ HTTP based protocol stacks for MMSM implementation." Ex-1010, 22. Therefore, in my opinion, once registered, application-specific MMs may be delivered by the Relay/Server using a protocol that includes TLS to the registered application. Additionally, because a secure message link is used for the communication between the MMS Relay/Server and the MMS User Agent (as described with reference to 1[a]/15[a], which I incorporate here), a POSITA would have understood that the messages including the application data are "secure message link messages."

169. A POSITA would have found it obvious, in view of TS-23.140's disclosures, that the above-described procedure of delivery of application-specific data to an application could be performed for multiple applications that are intended to send or receive application-specific data. For example, TS-23.140 provides examples of applications (e.g., chess application, messaging/discussion applications, news applications, weather applications) that use MMS for application specific messaging. *See, e.g.*, Ex-1004, p. 55 ("The additional application/implementation specific control information shall be used for all future needs that are not supported

by the application identifier of the destination application and the identifier of the originating application, such as specifying a particular logical channel in the application addressing method (e.g., 'discussion thread #05') or distinguishing between multiple instances of the same application (e.g., 'chess application #02'); p. 14 (explaining that VAS application are "applications providing value added services (e.g. news service or weather forecasts) to MMS users"), pp. 54-55 ("Applications that … transport application specific data using MMS … reside on an MMS User Agent…. Applications that want to transport data specific to applications other than MMS will initially need to register with the appropriate MMS User Agent"; describing different types of applications, e.g., chess and messaging applications"). Indeed, MMS provided a well-established infrastructure for application messaging and as described above, applications would naturally use MMS messaging to facilitate mobile content delivery (e.g., delivery of news, weather, or application related messaging).

170. Further still, by 2009, it was well-known for mobile/MMS-capable user devices/terminals to include multiple applications, and for such applications to send and receive application-specific data using MMS. For example, Mostafa discloses and depicts (below) multiple applications, e.g., chess, download, and location applications, that interface with the MMS User Agent (UA) to facilitate sending and receiving of application data via MMS. *See* Ex-1036, 3-4 (pp. 732-33, Figs. 3-5).

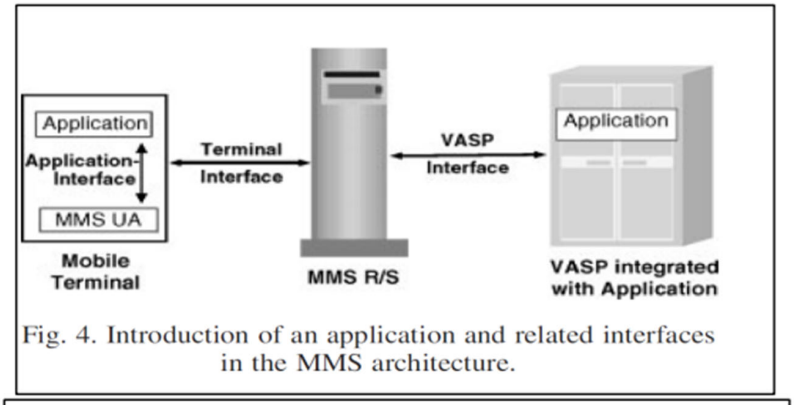


Fig. 4. Introduction of an application and related interfaces in the MMS architecture.

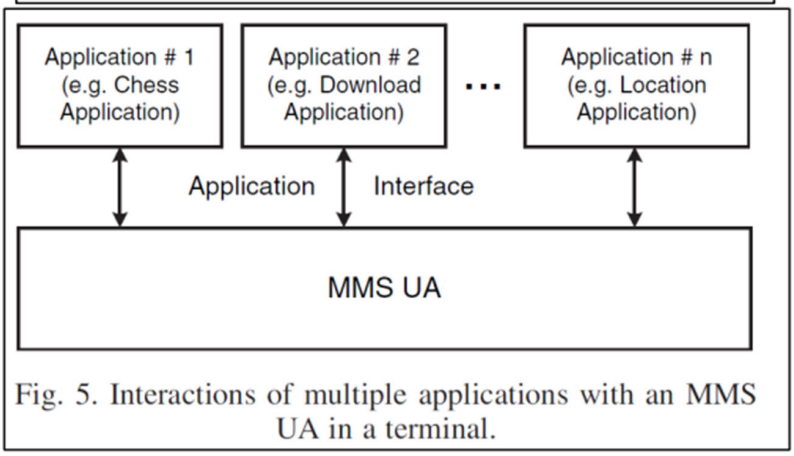


Fig. 5. Interactions of multiple applications with an MMS UA in a terminal.

See also id., 5 ("As for example, a chess application within a terminal may offer its user to play chess with a user of a different terminal having the same application. If both the terminals are also MMS-capable, the application may use an MM to convey the information about each move made by a user on the chessboard to the other user. Other interactive games (e.g., golf, snooker) in a mobile terminal can also use MMS as the transport similarly. A location service could also be based on this new MMS feature, so that a location server, acting as an application over a VASP, can provide location specific information to a terminal. Moreover, the feature is also useful for downloading any content (e.g., a media object, a new game, a new plug-in of an

existing game, an application, media codec) from a VASP. Furthermore, a configuration application could use the feature to get a new or an updated configuration of a terminal (e.g., connectivity information for any network service like MMS, browsing) from a network. These are just few examples, and the feature can be used for many other purposes.").

171. Thus, a POSITA would have understood or found it obvious that the UEs (i.e., the wireless end-user devices) would include multiple applications (i.e., software components) that are capable of receiving and processing the registered application data received from the MMS Server/Relay.

172. For the above reasons, TS-23.140 discloses or renders obvious 1[b]/15[b].

d. ***1[c1]****: an interface to a network to receive network element messages from a plurality of network elements, /* ***15[c1]****: receiving network element messages from a plurality of network elements,*

173. TS-23.140 discloses or renders obvious an interface (e.g., including MM1, MM3, MM7) to a network to receive network element messages (e.g., messages including application-specific data and other addressing data) from a plurality of network elements (e.g., MMS User Agents resident on MMS client devices, MMS VAS Applications).

174. In more detail, the following figure shows an MMS network environment in which communications are facilitated between various MMS User

Agents, the MMS Server/Relay, MMS VAS Applications, and one or more external servers over various networks.

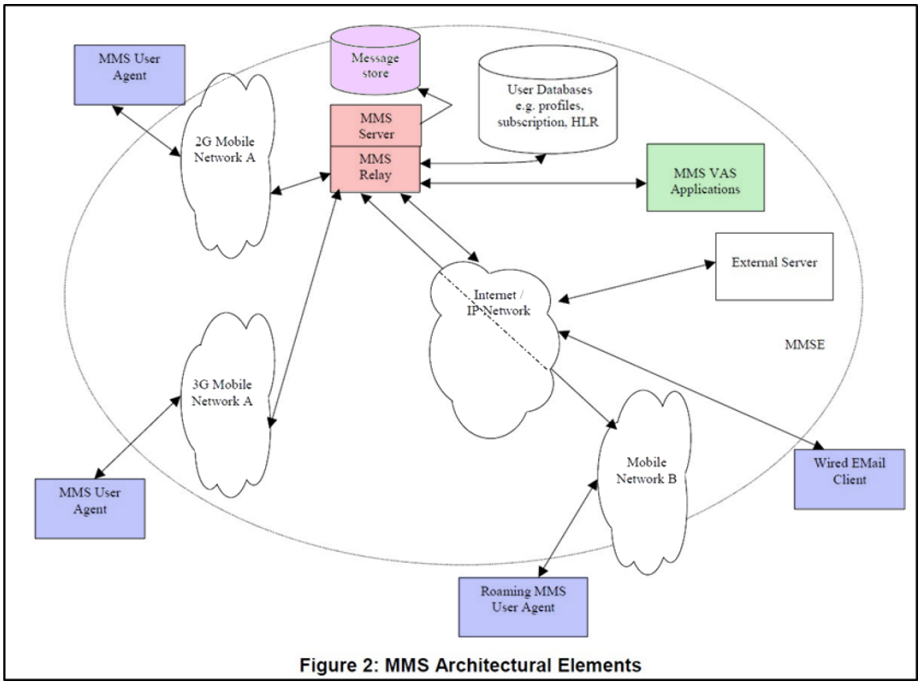


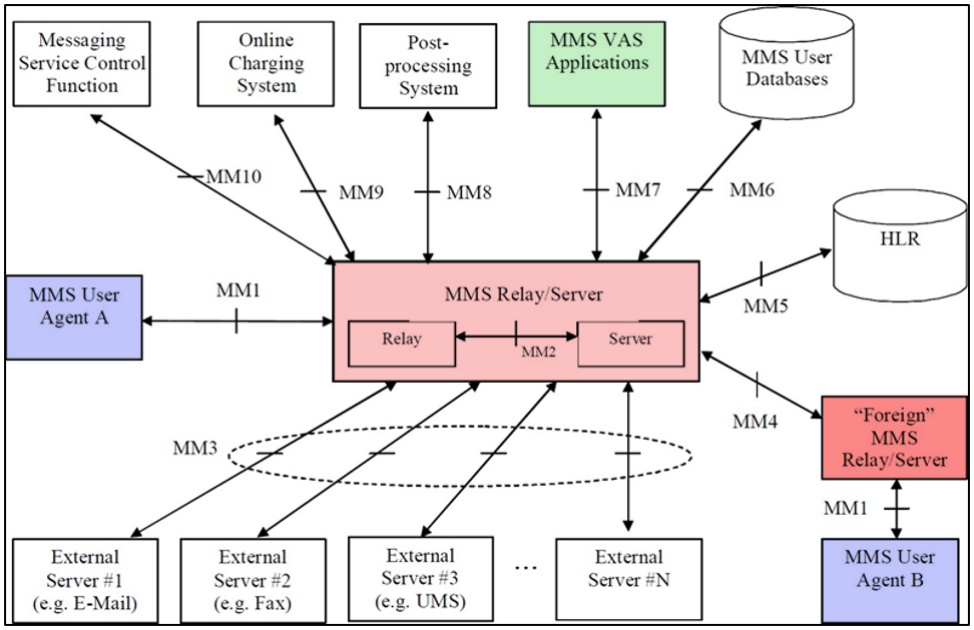
Figure 2: MMS Architectural Elements

Ex-1004, § 4.2, Fig. 2, p. 17 (describing the above MMS environment as "***a collection of MMS-specific network elements***.").

175. And, as described with reference to 1[a]/15[a] and 1[b]/15[b], the MMS Relay/Server receives messages from one or more of these network elements (i.e., network element messages), such as one or more MMS User Agents and MMS VAS Applications.

176. TS-23.140 further discloses various "interfaces" facilitating network communication of such network element messages. Ex-1004, §6.1, pp. 23-24. For example, the following figure shows the various interfaces facilitating

communications between MMS User Agents, MMS VAS Applications, and MMS Relay/Server, including the MM1, MM2, MM7, and MM4 interfaces:



Ex-1004, §6.1, Fig. 3, pp. 23-24.

177. Specifically, TS-23.140 discloses that (1) the MM1 interface is the "reference point between the MMS User Agent and the MMS Relay/Server," (2) the MM4 interface is the "reference point between the MMS Relay/Server and another MMS Relay/Server that is within another MMSE," and (3) MM7 interface is "the reference point between the MMS Relay/Server and MMS VAS Applications." Ex-1004, §6.1, pp. 23-24.

178. As shown in the above Figures, the interfaces are to various networks, including 2G/3G mobile networks, and IP/internet networks. Ex-1004, Fig. 2.

179. Additionally, as described with reference to 1[b]/15[b], applications that are registered with MMS User Agents or MMS VAS application can transmit application data to applications registered with other MMS user agents and VAS applications, in which case, the communications are facilitated via the MMS Server/Relay. *See* Ex-1004, §7.1.18, pp. 54-55 (contemplating delivery of application specification data to an application via the user agent, using that application's identifier).

180. For the above reasons, TS-23.140 discloses or renders obvious 1[c1]/15[c1].

> *e.* ***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; /* ***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;*

181. TS-23.140 discloses or renders obvious that the received network element messages (e.g., abstract messages received from applications resident on the MMS User Agent or the MMS VAS Applications) comprise respective message content, including data for (e.g., application data and/or other control information) and identification of a respective one of the authorized software (e.g., identification

of destination application(s)), and requests for delivery of the respective message content to respective wireless end-user devices.

182. In more detail, and as described with reference to 1[a]/15[a] and 1[b]/15[b], TS-23.140 contemplates use of MMS to facilitate transport of application data from one device (e.g., server or terminal) and its associated agent (MMS User Agent or MMS VAS application) to another device (server or terminal) and its associated agent (MMS User Agent or MMS VAS Application). Ex-1004, §7.1.18.1, pp. 54-55.

183. Specifically, TS-23.140 discloses that the transmission of application data is facilitated by an originating application triggering the MMS User Agent or MMS VAS Application to which it is registered to send an abstract message (that includes this application data) to a destination application. Ex-1004, §7.1.18.1, pp. 54-55 ("Based on the negotiated details upon application registration process ***an application may trigger an MMS User Agent or an MMS VAS Application to submit certain abstract messages***. Upon triggering an MMS User Agent or an MMS VAS Application to send an abstract message ***the MMS User Agent or MMS VAS Application may receive information from the application***. The MMS User Agent or MMS VAS Application ***may insert this information in both the information elements and/or payload (if present) of the abstract message***."), pp. 55-56 (application data can include "application/implementation specific control

information"); §7.1.18, pp. 54-55, §3.1, p. 14 (defining abstract message as "information which is transferred between two MMS entities used to convey an MM and/or associated control information between these two entities").

184. The abstract message further includes the destination application's application identifier. *Id.*, 54-55 ("***Abstract messages that are sent by an MMS User Agent*** or an MMS VAS Application on behalf of an originating application ***shall contain a destination application identifier***."). The originator also "indicate[s]" the message recipient's address in the message. *Id.*, 26, 90, 190 (including "Recipient address" in messages sent over MM1). MMS Relay/Server receives this message and passes "on the destination application identifier" and "application data" to, e.g., another MMS User Agent. *See id.*, 54-56.

185. Additionally, given that abstract messages are sent by an originating MMS User Agent to MMS Relay/Server for delivery to an MMS User Agent with which the destination application is registered, a POSITA would have understood or found it obvious that the transmitted message comprises a request for delivery of the respective message content to a respective wireless end-user device (including the destination application resident on that device). Ex-1004, §7.1.18, pp. 54-55 ("If the MMS Relay/Server finds from the recipient MMS User Agent's capability indication (see clause 7.1.3.1) that the recipient MMS User Agent supports transport of application data, ***the MMS Relay/Server . . . shall pass on the destination***

application identifier, the 'reply-path' identifier (if present) and the additional application/implementation specific control information (if present) unaltered. Upon reception of an abstract message containing a destination application identifier (it can either be the ***MM1_notification.REQ, MM1_retrieve.RES or MM7_deliver.REQ transactions***), the receiving MMS User Agent or MMS VAS Application shall first check if the destination application resides on it."); Fig. 6 (showing typical message exchange between MMS User Agents via MMS Server/Relay;); Ex-1004, §7.1.2.1, p. 28; Ex-1004, § 7.1.3, p. 29; *see also* 1[c1] above (further explaining and describing receipt of network element messages, including above-described messages exchanged between applications).

186. A POSITA would have understood or found it obvious that the recipient/destination application would be an authorized/registered application (*see* 1[b]) because applications intending to send/receive application data "need to register with the appropriate MMS User Agent" using their "application identification value" and then, upon message receipt, the recipient application would then need to be identified by MMS User Agent using its identification value as being resident on the user device before the message is transmitted to this application. *See* Ex-1004, 54-56.

187. Moreover, a POSITA would have understood in view of TS-23.140's disclosures (see § VII.A *supra* and 1[pre]-1[b] discussion above) MMS Relay/Server

can interface with multiple devices and their respective MMS User Agents, such that it transmits messages to multiple MMS User Agents/devices for delivery to their respective applications). As such, a POSITA would have understood or found it obvious that MMS Relay/Server receives requests for delivery of message content to multiple wireless end-user devices in the MMS environment.

188. For the above reasons, TS-23.140 discloses or renders obvious 1[c2]/15[c2].

> *f.* ***1[d1]-1[d2]****: a message buffer system including a memory and logic, 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, /* ***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;*

189. TS-23.140 discloses or renders obvious a message buffer system including a memory that buffers content from the received network element messages (e.g., message store for storing the received message from, e.g., an application associated with the originating MMS User Agent or VAS application, for which delivery is requested to a given one of the wireless end-user devices (e.g., an application running on a client device associated with the MMS User Agent for that device).

190. As described with reference to 1[c1]-1[c2] (which I have incorporated here), the network element messages are received from one or more network

elements by the MMS Relay/Server, and delivery of these messages is requested to an application resident on a user device (and associated with its MMS User Agent).

191. Moreover, the "***MMS Relay/Server is responsible for storage*** and handling ***of incoming and outgoing messages*** and for the transfer of messages between different messaging systems." Ex-1004, § 4.2, p. 17. Fig. 2 of TS.123.140 shows a message store coupled to the MMS Server/Relay:

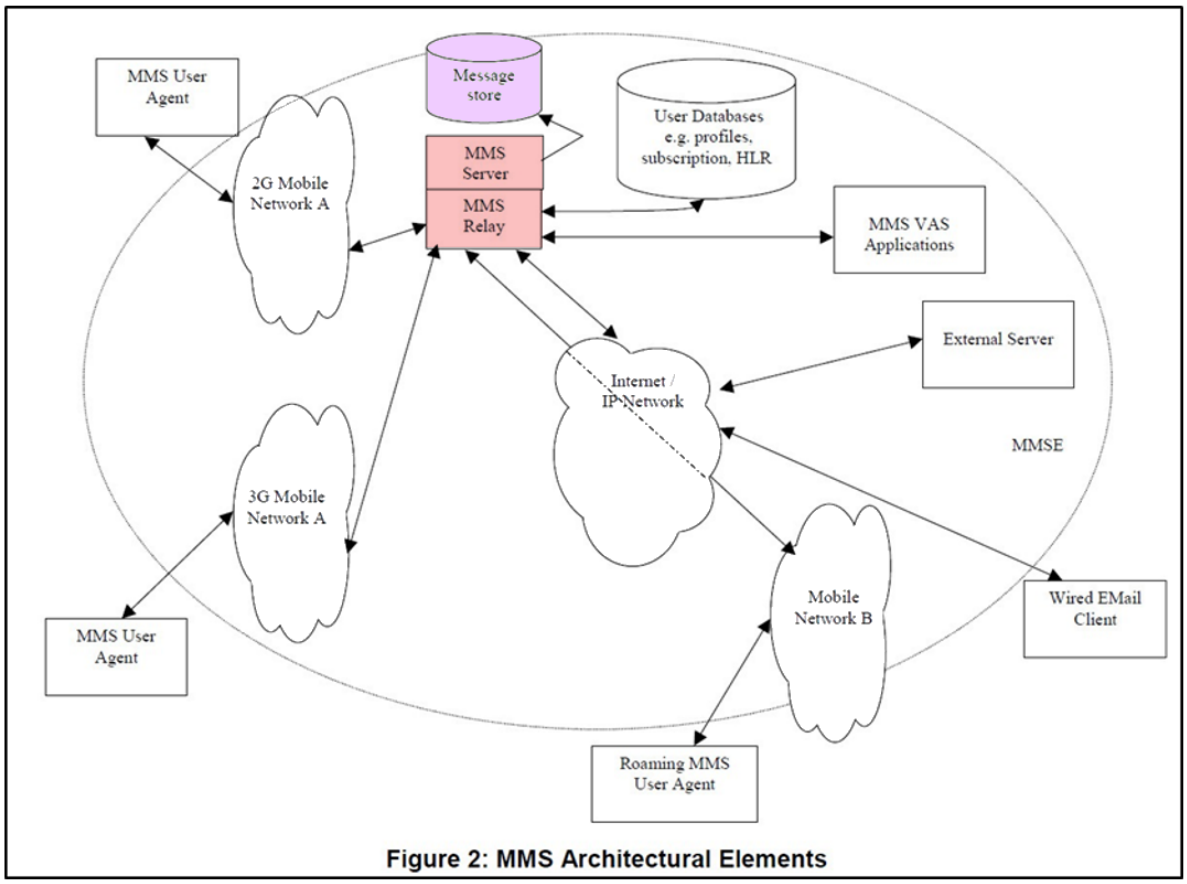


Figure 2: MMS Architectural Elements

See also Ex-1004, § 5.2, p. 21 (describing "temporary storage of messages" received and send by MMS User Agents; facilitating transport of application data).

192. "Upon reception of an MM from an originator MMS User Agent," the "originator MMS Relay/Server . . . ***shall retain the MM until the earliest desired time of delivery***, if the optional feature of earliest time of delivery is supported by the originator MMS Relay/Server. . . ." Ex-1004, § 7.1.1, 7.1.2, pp. 26-28.

193. Similarly, "***[u]pon reception of an MM the recipient MMS Relay/Server*** [which can be the same as the originator MMS Relay/Server]," this MMS Relay/Server can (1) ***store the MM at least until one of*** "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," and (2) "shall generate a notification to the recipient MMS User Agent." *Id.*

194. TS-23.140 further describes "persistent storage of messages" in a "Persistent Network-Based Storage" (called MMBox) ***which is associated with "the MMS Relay/Server and into which Multimedia Messages (MMs) may be stored, retrieved, and deleted***." Ex-1004, § 5.2.1, pp. 21-22. As such, when a multimedia message is received from an MMS User Agent, the MMS Relay/Server "***may store the MM into an MMBox*** . . ." *Id.*, pp. 26-28.

195. Additionally, as described with reference to 1[d3]-1[d4] (which I incorporate here), the MMS Relay/Server includes a message buffer system with logic that facilitates delivery of messages upon occurrence of one or more message delivery triggers.

196. For the above reasons, TS-23.140 discloses or renders obvious 1[d1]-1[d2]/15[d1].

> *g.* ***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 /* ***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*

197. As explained below, TS-23.140 discloses or renders obvious the message buffer system including logic that determines when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred (e.g., user request for a message, expiration of a message expiry period, determination of user settings or recommendations specifying retrieval mode to use for received message, conformity of message to MMS User Agent specification of a size restriction).

198. Additionally, and as further explained below, TS-23.140 renders obvious that the receipt of a message is not a message delivery trigger and one of the

message delivery triggers is an occurrence of an asynchronous event with time-critical messaging needs (e.g., a user request for a received message).

199. In more detail, TS-23.140 discloses that messages are delivered from the MMS Relay/Server to the MMS User Agent when the MMS Relay/Server sends a notification, and the MMS User Agent responds with a retrieval request for delivery. *See, e.g.*, Ex-1004, 66-69.

Table 7: abstract messages for notification of MM in MMS

Abstract message	Type	Direction
MM1_notification.REQ	Request	MMS Relay/Server -> MMS UA
MM1_notification.RES	Response	MMS UA -> MMS Relay/Server

However, TS-23.140 also specifies several conditions to trigger the MMS User Agent to issue the retrieval request and secure delivery.

200. If the claimed "trigger" must be a condition after which the next step is an attempt to deliver the message (e.g., the last traffic light before a destination), TS-23.140 discloses:

(1) When the recipient User Agent is configured for manual retrieval, the User Agent manually issues (trigger) a retrieval request in response to a notification to initiate a delivery attempt. Ex-1004, 28 (§7.1.2); *id.*, 28-29 (§7.1.2.1).

(2) When the recipient User Agent is configured for automatic delivery, the User Agent automatically issues (trigger) a retrieval request in response

to a notification to initiate a delivery attempt. Ex-1004, 28 (§7.1.2); *id.*, 28-29 (§7.1.2.1).

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

If, however, a "trigger" can include conditions that starts or restarts an attempted delivery (e.g., any traffic light on the way to a destination), in addition to the example triggers (1) and (3) I identify above, the following are also triggers:

(4) When the MMS Relay/Server receives a message without a specified delivery time (trigger), it immediately initiates the notification/retrieval process for a delivery attempt. Ex-1004, 26-27 (§7.1.1).

(5) When the MMS 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. EX-1004, 27 (§7.1.1).

(6) When the MMS Relay/Server receives a retrieval request from the recipient User Agent with a request for deferred delivery, the expiration of a timer or event occurrence at the delivery time (trigger) initiates a delivery attempt from the recipient User Agent and determines that a

specified deferred-delivery period is met or event occurrence at the delivery time, triggering an attempted delivery. *See id.*, 28-29 (§7.1.2.1).

But if a "trigger" can include conditions imposed on the messages controlling whether a message will be delivered at all (e.g., any traffic lights ***plus*** road restrictions, like maximum height/weight to get to a destination), in addition to (1)-(6), the following is another trigger:

(7) When the MMS Relay/Server receives a retrieval request from the recipient User Agent with a "size restriction," a determination that the message conforms to that restriction (trigger) initiates a delivery attempt. *Id*., 29-30 (§7.1.3).

201. Under any possible interpretation, because the MMS Relay/Server must monitor when these events occur or conditions are satisfied, in my opinion, a POSITA would have understood or found it obvious that the MMS Relay/Server includes logic configured to determine when they have occurred for a given UE/mobile device.

202. Additionally, under any possible interpretation, in my opinion, a POSITA would have understood or found it obvious that, for at least some received network element messages, message receipt by the memory store/MMBox alone is not a trigger. Specifically, if a trigger is a condition that controls whether an

attempted message delivery is made (conditions (1)-(7)) or is a condition that starts/restarts an attempted delivery (the conditions in (1), (3)-(6)), the receipt of an message without an earliest desired time of delivery (condition (4)) is a trigger. However, the receipt of message with an earliest desired time of delivery (at least some of the received network element messages) is not itself a trigger; attempted delivery is only triggered by a determination that the earliest desired time of delivery condition is met (conditions (5) or (6)). And if a trigger is a condition after which the immediate next step is an attempt to deliver the message (conditions (1)-(3)), the receipt of the MM—with or without a desired time of delivery—is not a trigger. Under this interpretation, none of the received messages are triggers, which satisfies the requirement that at least some of the received network element messages are not triggers.

203. Further, under any possible interpretation, a POSITA would have understood or found it obvious that at least some of the triggers occur upon an asynchronous event with time-critical messaging needs. For instance, unless a trigger must be a condition after which the immediate next step is an attempt to deliver the message, the receipt of the message in (4) triggers message delivery immediately (i.e., an asynchronous event with time-critical messaging needs). And under any possible interpretation, the manual retrieval in (1) is a trigger occurring upon an asynchronous event with time-critical messaging needs because the user

must manually cause the MMS User Agent to request delivery. EX-1004, 20. The ’192 Patent itself discloses that a manual “user request” for delivery constitutes such a message-delivery trigger. *See* Ex-1001 (’192 Patent), 38:50-63.

204. In my opinion the “triggers” I have identified above are consistent with the ’192 Patent’s description of what constitutes a trigger. For instance, the ’192 Patent explains: “the service control server link 1638 can determine when to transmit based potentially on several parameters, such as one or more of: periodic timer trigger, waiting until a certain amount of service usage or traffic usage has occurred, responding to a service agent message, responding to a service agent request, initiated by one or more servers, initiated by a verification error condition, and/or initiated by some other error condition.” Ex-1001 (’192 Patent), 70:6-13. The ’192 Patent further discloses: “In addition, the service control server link 1638 can provide for an efficient communication link based on various embodiments related to the timing of transmissions over the service control link, as similarly discussed above with respect to the service control device link 1691 description. For example, the timing functions, such as asynchronous messages or polling for messages, constant frequency transmission, transmission based on how much service usage or data traffic usage has taken place, transmission in response to device side control link message, service verification error events, other error events, and/or other message transmission trigger criteria can be determined, controlled and/or initiated

by either the device side or the network side depending on the embodiment." Ex-1001 ('192 Patent), 70:17-30.

205. Additionally, the '192 Patent makes clear that any component on the network (such as the server) and the (message-receiving) device can initiate the trigger. *See, e.g.*, Ex-1001 ('192 Patent), 70:17-30 ("For example, the timing functions, such as asynchronous messages or polling for messages, constant frequency transmission, transmission based on how much service usage or data traffic usage has taken place, transmission in response to device side control link message, service verification error events, other error events, and/or other message transmission trigger criteria can be determined, controlled and/or initiated by either the device side or the network side depending on the embodiment.").

206. The '192 Patent discloses additional triggers for "the service control device link 1691," which are similar to the list of transmission/message delivery triggers described above. More specifically, the '192 specification states: "the service control device link 1691 determines when to transmit based potentially on several parameters including, for example, one or more of the following parameters: periodic timer trigger, waiting until a certain amount of service usage or traffic usage has occurred, responding to a service controller message, responding to a service controller request, initiated by one or more agents, initiated by a verification error

condition, initiated by some other error or status condition." Ex-1001 ('192 Patent), 38:30-38.

207. Other identified examples include a "transmission trigger [that] is based on waiting for an amount of service usage, and also including a minimum transmission rate that triggers a transmission according to one or more of the following parameters: a maximum time between transmissions clock to keep the service processor 115 in communication with the service controller 122 when little or no service usage is occurring, a polling request of some kind from the service controller 122, a response to a service controller heartbeat, a transmission generated by a service verification error event, or a transmission generated by some other asynchronous event with time critical service processor 115 (or service controller 122) messaging needs, such as a transaction or service billing event or a user request." *Id*., 38:50-63.

208. The '192 specification provides additional details of the "service usage" or "service activity trigger," noting that it can be "based on some other measure than traffic usage, such as a number of messages transacted, one or more billing events, number of files downloaded, number of applications run or time that an application has been running, usage of one or more specified applications, GPS coordinate changes, roaming event, an event related to another network connection to the device and/or other service related measures." *Id*., 39:11-19.

209. For the above reasons, TS-23.140 discloses or renders obvious 1[d3]/15[d2].

> *h.* ***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. /* ***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.*

210. TS-23.140 discloses or renders obvious that, upon determining that one of the message delivery triggers has occurred, the message buffer system includes logic for supplying one or more messages comprising the buffered content (e.g., stored message in the message store or MMSBox) to the transport services stack for delivery on the secure message link maintained between the transport services stack and a device link agent (e.g., MMS User Agent) on the given one of the wireless end-user devices.

211. As described in 1[d1]-1[d3] (which I incorporate here), TS-23.140 contemplates delivery of the stored message (in MMBox or another temporary storage) from the MMS Relay/Server upon the occurrence of one or more of the above-described delivery triggers. *See*, *e.g.*, Ex-1004, pp. 28-31, §7.1.3, 7.1.2.1. As I also described in 1[d3]/15[d2] (which I incorporate here), in my opinion, a POSITA

would have found it obvious that MMS Relay/Server includes logic for performing message delivery upon detecting occurrence of one or more of the above-described delivery triggers.

212. Additionally, as described in 1[a] (which I incorporate here), TS-23.140 discloses message delivery via the MMS Relay/Server's transport services stack (including the MM1 Transfer Protocol) that maintains secure message link (e.g., TCP/IP and TLS-based link) to a device link agent (e.g., a recipient MMS user agent) of a particular wireless end-user device (e.g., a UE/terminal corresponding to the recipient MMS User Agent). *See, e.g.*, Ex-1004, §6.2, p. 24, Fig. 4 (providing protocol framework to provide MMS; listing MM1 Transfer Protocol as part of MMS Relay/Server); Ex-1010, p. 22 (describing TLS-based link for data transmission between MMS User Agent and MMS Proxy-Rely in architectural configurations that employ HTTP based protocol stacks for MMSM implementation).

213. For the above reasons, TS-23.140 discloses or renders obvious 1[d4]/15[d3].

2. Claim 5

214. TS-23.140 renders obvious the message link server (as described with reference to claims 1 and 15 above) and further renders obvious 5[a]-5[c], as described below.

a. ***5[a]****: 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,*

215. TS-23.140 discloses or renders obvious that 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 (e.g., one or more MMS User Agents) from at least a subset of the device software components (e.g., one or more applications).

216. In more detail, TS-23.140 discloses that an application executing on a user device/UE (i.e., a device software component) can forward a message including application data (i.e., upload message) to the UE's MMS User Agent (i.e., the device link agent), which in turn can submit the message to the MMS Relay/Server. Ex-1004, §7.1.18, pp. 54-55; § 3.1, p. 14. TS-23.140 also discloses that the MMS User Agent can submit/forward to the MMS Relay/Server MMs with requests to store them. EX-1004, 40.

217. A POSITA would have understood that an application receiving application data using MMS can also send application data to an application resident on another device (e.g., another device with an MMS User Agent or an MMS VAS Application).

218. For example, TS-23.140 discloses that "an application may trigger an MMS User Agent or an MMS VAS Application to submit certain abstract messages. Upon triggering an MMS User Agent or an MMS VAS Application to send an abstract message the MMS User Agent or MMS VAS Application may receive information from the application. The MMS User Agent or MMS VAS Application may insert this information in both the information elements and/or payload (if present) of the abstract message." *Id.*, 55.

219. Both the MMS User Agent and the MMS Relay/Server facilitate "transport of application data." *Id.*, §§5.1.1 and 5.1.2, pp. 19-21, 35 (describing transport of application data by MMS User Agent to another MMS User Agent via the MMS Relay/Server), 55 (transporting data between applications of two MMS User Agents). Moreover, as described with reference to 1[a]/15[a] (which I incorporate here), the MMS Relay/Server includes the MM1 Transfer Protocol that facilitates receipt/transport of application data from the MMS capable UE/MS to the MMS Relay/Server. Ex-1004, §6.2, p. 24, Fig. 4. From these disclosures, a POSITA would have understood or found it obvious that the transport stack of the MMS Relay/Server receives message including application data from an application via a particular MMS User Agent.

220. Moreover, in the MMS Environment contemplated by TS-23.140 (*see* Figs. 1 and 2), a POSITA would have understood that multiple different devices,

including end-user devices/terminal, would be able to use an MMS Relay/Server for transmission of data to multiple other devices/server/network elements.

221. In such an environment, a POSITA would have found it obvious for the MMS Relay/Server to receive upload messages (including application data) from multiple MMS User Agents and their respective applications requesting delivery of such data, and such communications would happen over the dedicated TLS-based links between each such MMS User Agent and the MMS Relay/Server (as described with reference to 1[a]-1[b]/15[a]-15[b]).

> *b.* ***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,*

222. TS-23.140 discloses or renders obvious each of the upload messages identifying a corresponding one of the network elements (e.g., recipient MMS User Agents on UEs) to which the device respective software component (e.g., originating application on a sending UE) has requested delivery.

223. As described in 5[a] above (which I incorporate here), the application data is included in the upload message (i.e., the abstract message) from an originating MMS Agent and its associated device, and this abstract message further includes the "application identifier of the destination application." Ex-1004, §7.1.18, pp. 55-56 ("***If the MMS Relay/Server finds from the recipient MMS User Agent's capability indication (see clause 7.1.3.1) that the recipient MMS User Agent***

supports transport of application data, the MMS Relay/Server . . . shall pass on the destination application identifier, the 'reply-path' identifier (if present) and the additional application/implementation specific control information (if present) unaltered."). Because the message includes an identifier of a destination application resident on a particular device/terminal, a POSITA would have understood that upload message identifies the corresponding device/terminal (i.e., the network element).

224. Additionally/alternatively, as with sending any multimedia messages (MMS), including MMs that include application data, the sender MMS User Agent "indicate[s] the address of the MM recipient." *See* Ex-1004, p. 26 (§ 7.1.1); *see also* p. 90, 190 (showing data types included in messages sent over MM1 interface between MMS User Agent and the MMS Relay/Server, including "Recipient address").

225. Indeed, because messages are transmitted from a MMS User Agent to a recipient MMS User Agent (or MMS VAS Application) (on another device/server), in my opinion, a POSITA would have found it obvious for the message to include an identifier or address for the device/server to which the message is being transmitted.

226. A POSITA would have understood or found it obvious that routing messages to particular devices would be facilitated when including some device

identification to facilitate that routing over a network. Indeed, TS-23.140 discloses that for MM1 communications (between MMS Relay/Server and MMS User Agents), the message must be addressed using three addresses: "the address of the MMS Relay/Server, the address of the recipient and the address of the originator." Ex-1004, p. 57. "The recipient's address can be a user's address, a user's terminal address, or a short code." *Id.* Additional details about message addressing and formats are provided below (*id.*):

227. The MMS User Agent and MMS Relay/Server shall support both E.164 (MSISDN) and RFC2822 addressing formats. The reference point MM1 should support a way to indicate the used address type to enable future extension. The encoding of the addressing is up to the corresponding MM1 implementation (cf. Annex B).

E.g. the originator MMS User Agent may specify each of the address fields in one of the following formats:

1) *RFC 2822 address (FQDN or unqualified)*

2) *PLMN address: ["+" | "*" | "#"] [digit | "*" | "#"] ... ["/TYPE= PLMN"]*

3) *Other "/TYPE= "*

The "/TYPE= " field specifies the address type. When PLMN format is used the type is optional. The "/TYPE= " convention provides flexibility for future enhancements.

When the "/TYPE=" qualifier is absent, the MMS Relay/Server should resolve potential ambiguities by applying the following logic to the address in the following order:

1. *if it contains the "@" character, the address should be interpreted as an FQDN RFC2822 address*
2. *if it is completely numeric, except possibly including "+", "*", or "#", it should be interpreted as "/TYPE= PLMN", e.g. an E.164 address, a local telephone number, or a numeric short code,*
3. *otherwise, it should be interpreted as an unqualified RFC2822 address (alphanumeric short code)*

228. Thus, a POSITA would have found it obvious to include such addressing in the upload message to facilitate transmission of data between devices of a network. Thus, given TS-23.140's disclosures and well-known device addressing aspects for network communications, a POSITA would have found it obvious to include such addressing in the upload message for transmitting data between network devices.

c. ***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.*

229. As I described in 1[a]-1[c1] above (and incorporate here), the communications between the MMS Relay/Server and the recipient MMS User

Agents would happen using the MM1 and MM4 interfaces, which facilitate the network communications between network elements via one or more MMS Relay/Servers, where the network elements would be the MMS User Agents residing on MMS client devices/UEs.[5] Other interfaces—such as MM2 and MM7—are also used to interface other network elements. Ex-1004, §6.1, pp. 23-24. These interfaces are also operable to work over a 2G, 3G, and IP/internet network. Ex-1004, §6.1, pp. 23-24.

230. For the above reasons, TS-23.140 discloses or renders obvious 5[a]-5[c].

3. ***Claim 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.*

231. As described below, TS-23.140 renders obvious that at least one of the one or more messages for delivery by the transport services stack comprises multiple identifier/data pairs.

232. As I described in 1[a], 1[c1]-1[c2], and 1[d3]-1[d4] (which I incorporate here), the MM1 Transfer Protocol is used to communicate between an MMS User Agent (or VAS Application) and an MMS Relay/Server, and an

[5] I note that claim 5 uses the term "network server system" that does not appear elsewhere in the claims. However, I have assumed that the "network server system" refers to at least the "message link server" that appears in claim 1.

originating MMS User Agent (or an originating MMS VAS Application) sends, via the MMS Relay/Server, application data as well as the identifier of the destination application to the destination application corresponding to a destination MMS User Agent. Thus, such a message includes an identifier/data pair in the form of the application data and the destination application identifier. Ex-1004, 54-56.

233. As I also explained in 1[c1]-1[c2], in my opinion, a POSITA would have understood that multiple registered applications on a UE/mobile device would receive messages including application data via the MMS Relay/Server. *See, e.g.*, EX-1004, 54-56; Ex-1036, 3-4.

234. In my opinion, a POSITA would have understood that, in some situations, application data/identifiers for multiple applications should be consolidated in a single message. For instance, TS-23.140 discloses that one delivery trigger is when the MMS User Agent becomes available/reachable (EX-1004, 28-30), meaning multiple messages may be queued for delivery at one time. In these circumstances, a POSITA would have been motivated to consolidate application data and corresponding application identifiers for multiple applications in a single message to achieve network efficiencies compared to transmitting multiple separate messages. *See, e.g.*, Ex-1039, Abstract (reducing number of messages signaled over a network "to reduce CPU utilization at both the source and the target system, which improves the overall system throughput and performance, and also reduces the

bottleneck at the network level."). A POSITA would have had a reasonable expectation of success in making this modification because it merely requires combining existing data structures.

235. Additionally, the concepts of broadcast (https://web.archive.org/web/20201128164630/https://csrc.nist.gov/glossary/term/broadcast) and multicast (Tom Bachert, IPv4 Multicast Security: A Network Perspective, SANS Institute, https://sansorg.egnyte.com/dl/rtTCC6TbTpFq) messaging are applicable, wherein a single address can be used to send messages to many or all devices. As such, a POSITA would have understood or found it obvious that there were multiple known techniques for delivering messages to multiple recipients (thereby providing messages with multiple identifier/data pairs that are intended for applications—where data for application is routed to that application using its identifier included in the message(s)).

236. For the above reasons, TS-23.140 discloses or renders obvious claim 6.

4. ***Claim 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.*

237. As described in 1[a] (which I incorporate here), MMS uses the MM1 Transport Protocols and TLS for secure messaging between the MMS User Agent and the MMS Relay/Server, and the MMS User Agent is responsible for sending messages by using such network protocols. TS-23.140's Relay/Server establishes

and maintains a link over an Internet network with User Agents executing on each user equipment ("UE")/mobile device. EX-1004, 17, 19-20. A POSITA would have found it obvious for the TLS-based communication link between the UE/client device and the MMS Relay/Server to be initiated by the client device.

238. Indeed, before the Critical Date, it was well-known for TLS-based communication between a UE/client device and a server to be initiated by the client device. *See, e.g.*, Ex-1012, 29:31-30:24 (describing that TLS requires a handshake phase and a data transfer phase, and in the handshake phase, the "client side initiates a TLS handshake by sending the server a ClientHello message"). But regardless, there were only two known options to choose from: (1) having the client initiate the link; or (2) having the server initiate the link, each of which would have been obvious choices from which to choose with a reasonable expectation of success.

239. Given that the MMS User Agent sends messages to the MMS Relay/Server (at the user terminal/client device) via a TLS-based connection and that TLS communications are initiated by a client device, a POSITA would have obvious that the MMS User Agent (i.e., the device messaging agent on at least one of the wireless end-user devices) initiates the respective secure Internet Data message link to the transport services stack of the MMS Proxy/Relay server.

240. Additionally, as described with reference to claim 1 (limitations 1[d2]-1[d4]), TS-23.140 discloses that MMS User Agent may be unavailable and

subsequently, when it becomes available, a POSITA would have found it obvious that the MMS User Agent may initiate and maintain a communication with the MMS Relay/Server to retrieve any messages stored therein and intended for MMS User Agent. Thus, this further confirms the obviousness of the MMS User Agent/device initiating the TLS-based secure Internet data message link to the MMS Relay/Server (and its associated network communication and transport protocol stock).

241. For the above reasons, TS-23.140 discloses or renders obvious claim 7.

5. ***Claim 9:*** *The message link server of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

242. TS 123.140 renders obvious that one of the message delivery triggers is the expiration of a periodic timer.

243. In more detail, TS-23.140 discloses that the originating MMS User Agent can set the desired time of delivery of the MM, in which case the MM is stored by the MMS Relay/Server until the ***earliest desired time of delivery***, before the message is delivered. *See* Ex-1004, § 7.1.1, p. 27.

244. In my opinion, a POSITA would have understood or found it obvious that specifying the desired time of delivery triggers a periodic timer upon the receipt of a message that expires when the delivery time is reached, which then triggers the delivery of the message. Such a timer would be periodic in that each time a message is received, the timer is triggered and continues for a particular period at the expiration of which, the message is delivered to the intended recipient.

245. Additionally, or alternatively, MMS User Agent uses periodic polling to identify new messages on an external server, and subsequently retrieves those messages—which also demonstrates another example a message delivery trigger that is the expiration of a periodic timer (as indicated by periodic polling). *See* Ex-1004, §8.3.3, pp. 90-91 (explaining that discovery of new messages on external servers can be done by "***periodic polling*** for messages on External Server, followed by retrieval by the MMS User Agent via the MMS/Relay Server"); §3.1, p. 14 (explaining that an external server is a "network entity/application of an external system such as Internet email, unified messaging system or facsimile to which MMs may be sent to and/or from which MMs may be received by an MMS User Agent via an MMS service provider").

246. A POSITA would have understood that (1) the above-described periodic polling for messages is akin to the claimed feature of a periodic timer expiring because the periodic polling would be done at periodic intervals (i.e., after a certain time period has elapsed/expired) and (2) this periodic polling is a message delivery trigger because, in response to receipt of such polling, messages from the respective servers would be provided/delivered to the MMS User Agent via the MMS Relay/Server.

247. For the above reasons, TS-23.140 discloses or renders obvious claim 9.

6. Claims 11-12: [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. ***[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.*

248. As described in 1[a], 1[d1]-[d2], and 1[d3] (which I incorporate here), TS-23.140 discloses that one of the message delivery triggers is a manual request for delivery from the MMS Relay/Server over a TLS-secured link (secure message link) in response to a notification from the MMS Relay/Server.

249. For the above reasons, TS-23.140 discloses or renders obvious claims 11-12.

7. Claim 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.*

250. As described in 1[d3] (which I incorporate here), TS-23.140 discloses that the message delivery triggers may include (among others) receipt of a message that does not specify an earliest desired time of delivery, receipt of a manual/automatic retrieval request, or receipt of a message indicating that the MMS User Agent has become available/reachable.

251. In this manner, a POSITA would have understood that TS-23.140 contemplates message delivery triggers that include receipt of certain types of

messages from one or more network elements (originating MMS User Agents operating of UEs/client devices).

252. For the above reasons, TS-23.140 discloses or renders obvious claim 13.

B. Ground 1B: Claims 2, 3, and 12 Are Rendered Obvious By TS-23.140 and Shen

1. ***Claim 2:*** *The message link server of claim 1, further comprising an encrypt function to encrypt the one or more messages supplied to the transport ser-vices 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.*

253. TS-23.140-Shen renders obvious 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.

254. As described with reference to 1[a] and 1[d4] (which I incorporate here), TS-23.140 renders obvious that messages are supplied to the MMS Relay/Server's transport services stack (the MM1 Protocol) for delivery on the secure message link (a TLS link) between the MMS Relay/Server and an MMS User Agent (resident on a particular user device/terminal). Although TS-23.140 discloses encrypting the connection between the MMS Relay/Server and MMS User Agent, it does not appear to disclose encrypting the messages themselves.

255. Shen, however, explains that the store and forward functionality of the MMS Relay/Server—without encrypting the messages—exposes a "potential security problem" given that such systems are "not an end-to-end solution …." Ex-1014, [0004].

256. To overcome this "security problem," Shen's invention in one embodiment discloses that "an authentication and key management module 502 is added in the MMS gateway to distribute symmetric keys to users and a certificate management module 504 is added in the user device," as shown below:

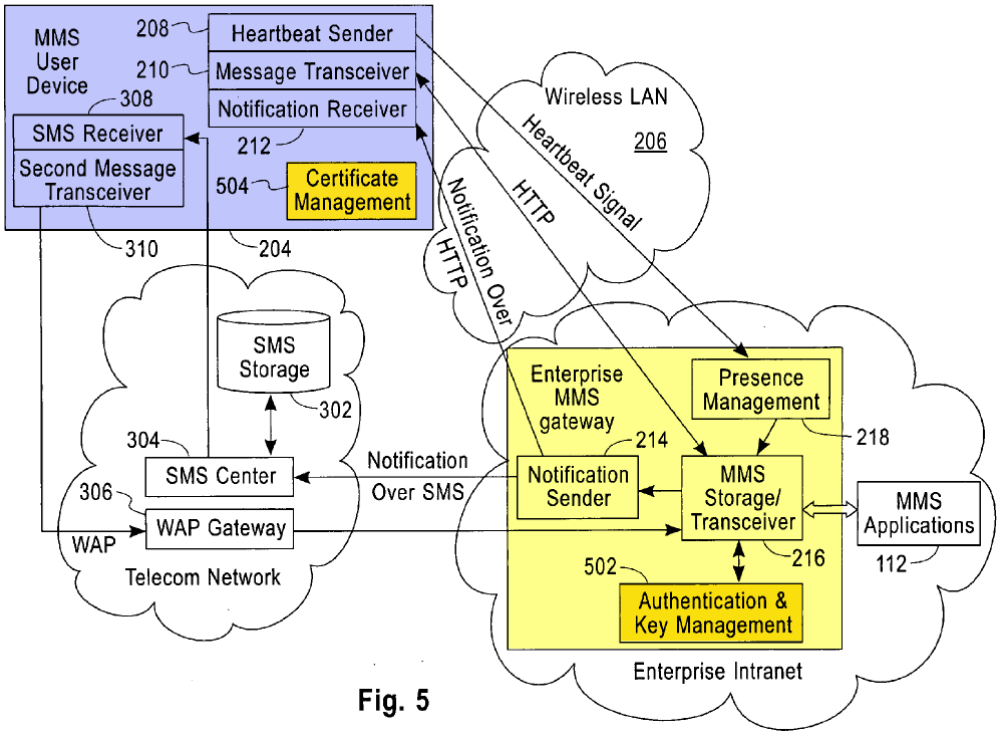


Fig. 5

Id., Fig. 5.

257. Using these modules, Shen discloses generation of a symmetric key that is then used by the MMS Relay/Server to encrypt data communications transmitted to the MMS User Device/Agent of user devices. *Id.*, [0055]-[0060]; *see also id.*, [0029]-[0034]. Shen also discloses that messages can be "encode[d] and decode[d] to protect the privacy of these messages." *Id.*, [0054].

258. In my opinion, a POSITA would have been motivated to combine the teachings of TS-23.140 and Shen to achieve an MMS system "having improved security," resulting from "provi[sion] [of] an end-to-end security solution for MMS applications," which is particularly beneficial in the context of "enterprise applications." Ex-1014, [0017], [0022] (describing additional advantages stemming from Shen's invention). Specifically, a POSITA would have been motivated to implement an "authentication and key management module 502" at the MMS Relay/Server that generates and "distribute[s] symmetric keys to users" and using this symmetric key to encrypt data transmitted to "the user device," as contemplated by Shen. *Id.*, [0054]-[0060].

259. Moreover, it was well known before the Critical Date for push serves to encrypt content of messages sent to push clients/client devices. Chou is one example of a push message system that uses a "WAP gateway" (e.g., a network message server) that encrypts content received from a "content server" for transmission to a "mobile client" that then "decodes the response." Ex-1009, 6:15-

46. Rakic and Shen provide additional examples of a push server encrypting message traffic. Ex-1015, [0065]-[0068]; Ex-1014, [0054]-[0060]

260. Combining the above-described teaching of Shen with those of TS-23.140 would have been obvious to a POSITA at least because such a combination would have merely involved combining prior art elements (an MMS network and message encryption) according to known methods (using encryption schemes and key generators) to yield predictable results (encrypted messages). Similarly, the combination would have amounted to using a known technique (message encryption) to improve similar devices (MMS networks) in the same way (securing content).

261. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification because Shen already envisions its proposal working in coordination with MMS-system architecture like that described in TS-23.140. Ex-1014, [0017]. Moreover, both TS-23.140 and Shen describe similar MMS environments and communications between MMS User Agents and the MMS Relay/Server, Ex-1004, pp. 16-18, 54-56; Ex-1014, [0029]-[0034], Figs. 1-2, 5, such that a POSITA would have found straightforward to modify TS-23.140's MMS Relay/Server to implement Shen's security modules.

262. Additionally, the resulting system's elements would each perform functions they were known to perform prior to the combination—i.e., TS-23.140's

system would continue to facilitate communication of messages between MMS Relay/Server and MMS User Agents, and Shen's teachings (as implemented in combination) would facilitate encryption of messages by the MMS Relay/Server that are then sent to the MMS User Agent.

263. Upon combining TS-23.140 and Shen, a POSITA would have appreciated that TS-23.140-Shen provides an encrypt function to encrypt messages supplied to the components implementing the MM1 Protocol for delivery on the TLS link maintained between the MMS Relay/Server and the UE/mobile device's MMS User Agent to "encrypt[] MMS messages with the shared secret key." Ex-1014, [0059].

264. For the above reasons, TS-23.140-Shen renders obvious claim 2.

> ***2.*** ***Claim 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.*

265. TS-23.140-Shen renders obvious 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.

266. As described with respect to claim 2 (which I incorporate here), TS-23.140-Shen renders obvious encrypting messages and transporting the encrypted messages to a particular MMS User Agent on a UE/terminal.

267. Additionally, as explained with reference to 1[a] (which I incorporate here), the combination renders obvious transporting these encrypted messages between the MMS Relay/Server and MMS User Agent using TLS encryption on the components implementing the MM1 Protocol.

268. For the above reasons, TS-23.140-Shen renders obvious claim 3.

3. ***Claim 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.*

269. TS-23.140-Shen renders obvious that the transmission is a heartbeat message generated by the given device link agent, or a request received from the given device link agent.

270. TS-23.140 discloses one trigger is when a recipient MMS User Agent becomes available/reachable. Ex-1004, 29.

271. Shen further teaches that a heartbeat message generated by a user's device is used to indicate reachability of the device to receive stored messages by pinging the gateway at intervals to express its reachability/availability. Ex-1014, [0033]-[0034].

272. In my opinion, a POSITA would have been motivated to incorporate Shen's "heartbeat" evaluation as an additional trigger in TS-23.140 by providing a specific mechanism (a heartbeat message) to trigger message delivery in a way that TS-23.140 already envisions (based on reachability of the MMS User Agent). For this reason, in my opinion, a POSITA would have been motivated to implement a heartbeat message generated by the MMS User Agent to ping the MMS Relay/Server in TS-23.140 at intervals because this would allow the MMS Relay/Server to periodically send buffered messages to the UE/mobile device when the device is available/reachable.

273. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification not only because Shen envisioned operating in an MMS network like TS-23.140 (which I described in claim 2 and incorporate here), but also because it merely involves configuring a messaging signal from the MMS User Agent to the MMS Relay/Server indicating the agent's availability to receive messages, and Shen already discloses what the contents of that message may be. Shen, [0034].

274. As described above, TS-23.140 renders claim 12 obvious.

C. Ground 1C: Claim 4 Is Rendered Obvious By TS-23.140 and Ellison

1. ***Claim 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.

275. TS-23.140-Ellison renders obvious executing the device link agent in a secure execution environment, with applications that interact with the device link agent operated outside of the secure execution environment.

276. TS-23.140 discloses both an MMS User Agent and applications executing on a UE/mobile device. EX-1004, 17-18, 23. TS-23.140 does not, however, appear to disclose implementing the MMS User Agent in a secure environment.

277. Ellison is just one example of prior art that uses secure environments to inhibit attacks from malicious software. Ellison, Abstract, 1:16-51. To inhibit attacks, Ellison discloses both an "isolated execution mode" (depicted in red below) where "access … is restricted" and a "normal execution mode" that "operates in a non-secure environment, or a normal environment" (depicted in green below) "without the security features provided by the isolated execution mode." Ex-1019, 4:65-5:1, 6:1-26, 8:25-32, Figs. 1A, 1C. Additionally, Ellison's teachings enable "normal" execution, for instance, for applications "***outside of the secure execution environment on that device***." *Id.*, 4:65-5:1, 6:1-26, Figs. 1A, 1C.

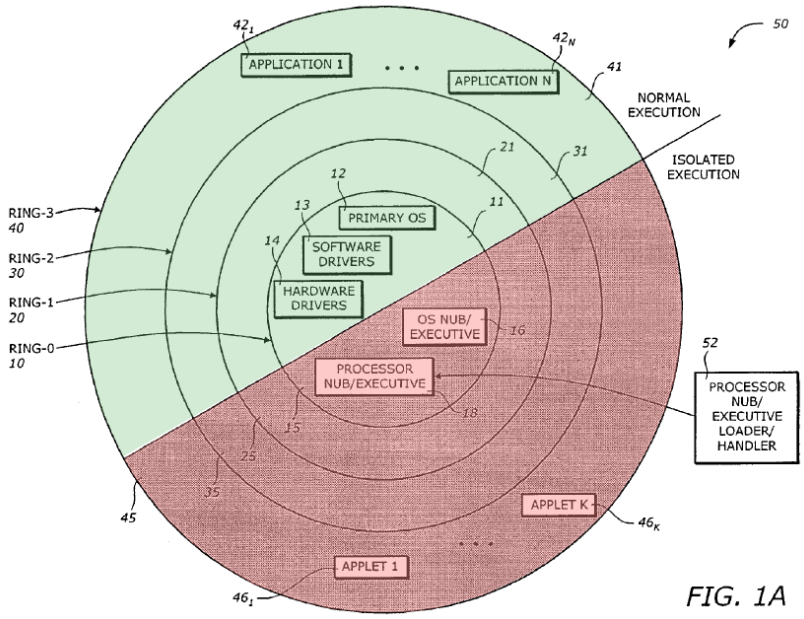


FIG. 1A

Ex-1019, Fig. 1A.

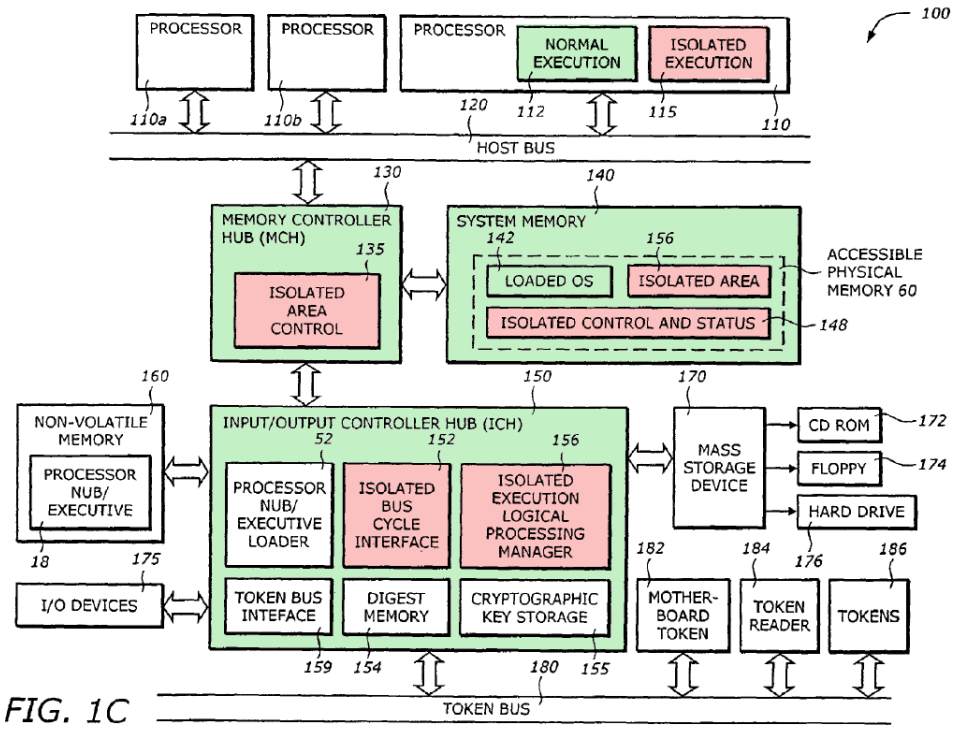


FIG. 1C

Id., Fig. 1C.

278. Ellison further discloses an "isolated area 70" which is "[a] memory area that is defined by the processor 110 when operating in [an] isolated execution mode." Ex-1019, 6:1-26, 8:25-32, Figs. 1A-1C, 2. Ellison refers to this technique, illustrated below in Figure 2, as a "secure platform." *Id.*, 6:1-26, 8:25-32, 8:66-9:62, Figs. 1A-1C, 2.

279. Ellison further teaches that the secure platform includes a "key generator 240" that "generates a key operating system nub key (OSNK) 203" which is then supplied to "trusted agents." *Id*., 8:66-9:6, Fig. 2. One example of a trusted agent is a "usage protector 250" that "uses the OSNK 203 to protect the usage of [a] subset 230." *Id*., 9:28-40, Fig. 2. Ellison's usage protector 250 uses a hashing function with the subset 203's ONSK 203 to determine if the subset 203 has been altered following changes (e.g., reads or writes to the subset 203), thereby providing "protection against unauthorized reads, and detection of intrusion, tampering or unauthorized modification" *Id*., 9:47-62.

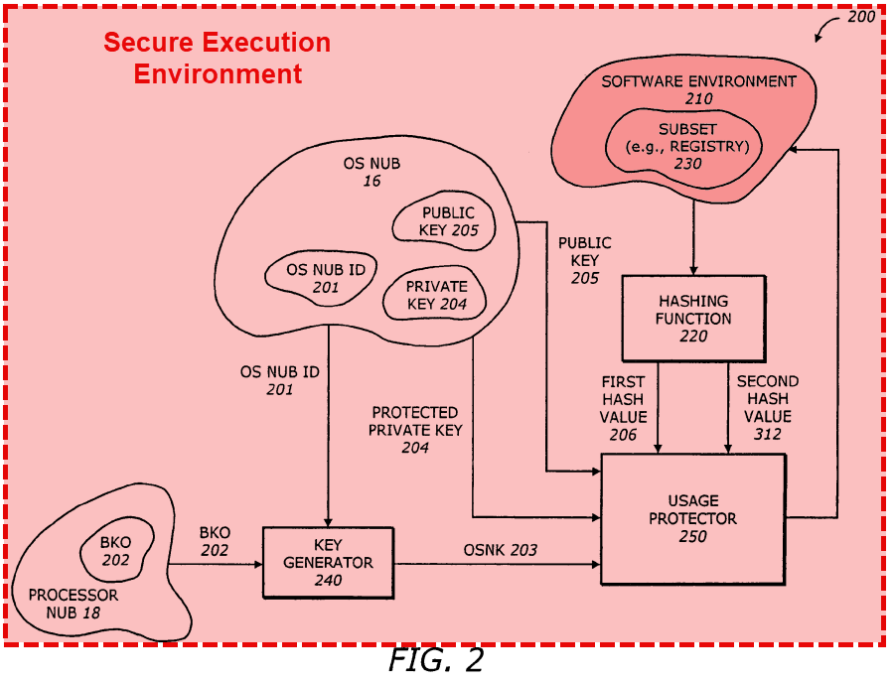


FIG. 2

Id., Fig. 2.

280. In my opinion, a POSITA would have found it obvious to combine these references such that Ellison's secure platform, which features hierarchical, nub-based normal and isolated execution environments, would have been incorporated into TS-23.140's UE/terminals. *Id.*, 8:25-32, 8:66-9:62, Fig. 2. A POSITA would have combined the teachings of these references for the following reasons.

281. First, a POSITA would have recognized that Ellison's secure platform would have improved the security of TS-23.140's MMS system by improving device security. *Id.*, 8:25-32, 8:66-9:62, Fig. 2.

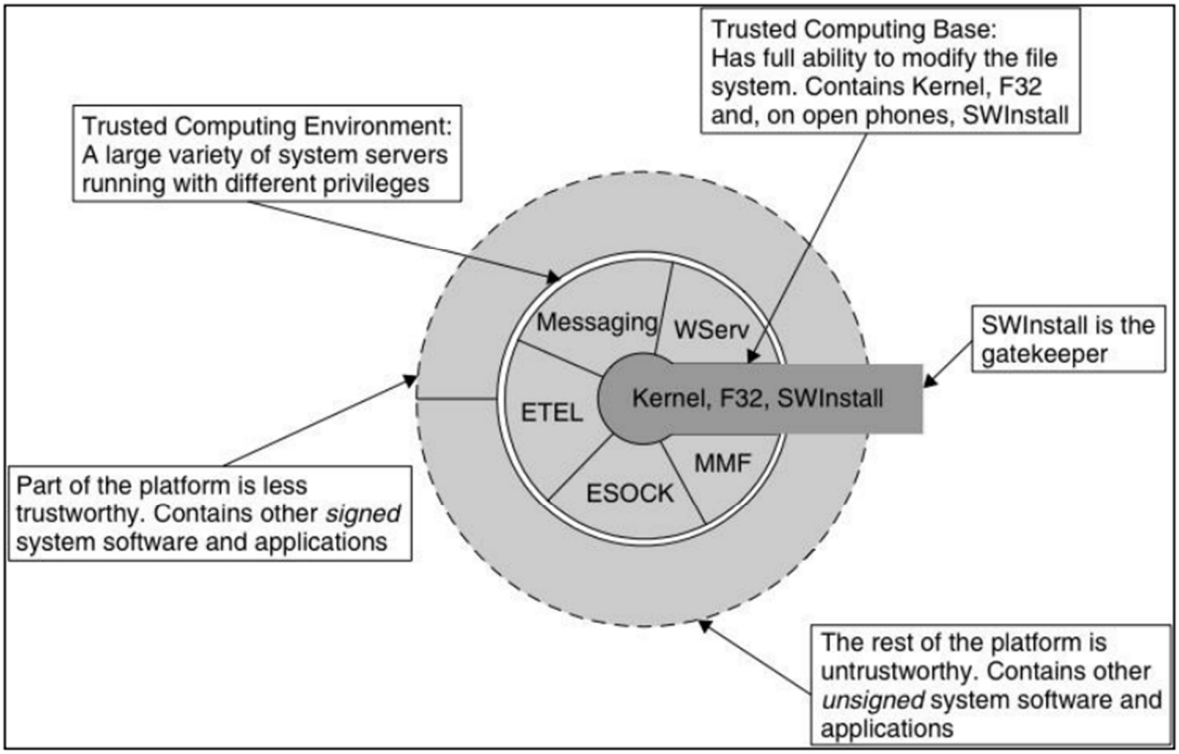
282. Second and relatedly, such increased confidence in device and/or network security provided by Ellison would have enabled a POSITA to TS-23.140's MMS services to additional UEs/servers (VAS Applications) *Id.*

283. Indeed, a POSITA would have understood that device level security improvements are an efficient, beneficial option for improving network security, which would have been a natural solution for TS-23.140's system, which contemplates secure/encrypted messaging in the MMS environments.

284. As explained below, combining the teachings of TS-23.140 and Ellison would have been obvious because such a combination would have merely involved combining prior art elements according to known methods to yield predictable results. Incorporating Ellison's security techniques into TS-23.140's mobile terminals would have been predictable and foreseeable with a reasonable expectation of success because Ellison describes that its techniques can be implemented in "computer system[s]" (such as TS-23.140's UEs) that would include a "processor." *Id.*, 5:11-16. Indeed, a POSITA would have recognized or found it obvious that UEs/terminals in TS-23.140 would include a "processor" capable of implementing Ellison's techniques.

285. Third, in the context of mobile devices and their associated operating systems, it was well-known before the Critical Date to implement more secure systems, with rings or tiers of protection around different device components. For

example, in the early 2000s, the Symbian mobile operating system (B. Li, et al., Symbian OS platform security model, LOGIN: Magazine (Aug. 2010) "Exhibit 1033"; Simon Higginson, Platform Security Concepts, in SYMBIAN OS PLATFORM SECURITY: SOFTWARE DEVELOPMENT USING THE SYMBIAN OS SECURITY ARCHITECTURE, 17-41 (Craig Heath ed., 2006)—Ex-1045) separated processes into three "Trust Tiers": The Trusted Computing Base (TCB), The Trusted Computing Environment (TCE), and Applications, as illustrated below:



Ex-1045, 24 (Fig. 1)

286. The first two layers/tiers shown were primarily dedicated to components of the Operating System with relatively higher operating privileged,

followed by the Applications tier with applications having relatively lower privileges. The first two tiers included the Trusted Computing Base and the Trusted Computing Environment. As shown, the Trusted Computing Environment includes, among other services, a Messaging service. The Application Tier could be further subdivided into Signed and Unsigned applications. Signed applications could interact with the TCE and have this layer execute higher privileged requests, whereas Unsigned applications could not access that same functionality.

287. Thus, it was well-known for operating systems to include the messaging services and clients/agents responsible for provision of those services, within the trusted computing environment, and for applications (including those that would be intended recipients of those messages) to be tiered/organized outside this environment. Given that TS-23.140 contemplates mobile terminals with an MMS User Agent that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging) from other applications/components of the environment, and Ellison's own disclosures confirming the same, a POSITA would have been motivated to implement Ellison's teaching of secure execution environment that enable such improved security within the MMS environment, such that the MMS User Agent is organized as part of the trusted computing environment

(i.e., the secure execution environment) and enables communications with applications residing outside this trusted/secure environment.

288. In an example of the combined system, Ellison's secure platform would have been incorporated into TS-23.140's mobile terminals, to secure the MMS User Agent, and further have a usage protector protecting messaging operations of the MMS User Agent. Ex-1019, 8:25-32, 8:66-9:62, Fig. 2. Specifically, Ellison's above-described teachings of a usage protector and secure platform, which operate within the secure, isolated area (secure execution environment) would be incorporated into TS-23.140's MMS User Agent executing on UEs/terminals, which in turn would provide secure access to applications residing outside the secure execution environment. Upon making this combination, a POSITA would have understood that TS-23.140's MMS User Agent would operate in an isolated, secure environment that "is protected by both the processor and chipset" (Ex-1019, 3:32-48), to reduce the MMS User Agent/user's messages exposure to attack. Applications that interact with the MMS User Agent would operate in normal or separate isolated environments to restrict the application(s) access to the MMS User Agent. *Id.*, 2:55-61 (noting applications operate in "Ring-3"—"the lowest level of the hierarchy" with the "least privilege").

289. A POSITA would have been motivated to so combine these teachings because (1) a POSITA would have recognized that Ellison's secure platform would

have improved the security of TS-23.140's MMS system by improving device security (*Id.*, 8:25-32, 8:66-9:62, Fig. 2); and (2) such device level security improvements are an efficient, beneficial option for improving network security, which would have been a natural solution for TS-23.140's system, which contemplates secure/encrypted messaging in the MMS environments. *Id*.

290. In other words, the combined system would enable more secure transmission of MMS messages to applications on the mobile terminal, and prevent potential bad actors from sending malware to the mobile terminal in the form of such messages or from modifying or otherwise gaining unauthorized access to the messages. *Id.*, 8:25-32, 8:66-9:62, Fig. 2.

291. Moreover, combining TS-23.140 and Ellison's teachings would have been obvious because such a combination would have merely involved combining prior art elements according to known methods to yield predictable results. Incorporating Ellison's security techniques into TS-23.140's mobile terminals would also have been predictable and foreseeable with a reasonable expectation of success because Ellison describes that its techniques can be implemented in "computer system[s]" (such as TS-23.140's UEs) that would include a "processor." *Id.*, 5:11-16. Indeed, a POSITA would have recognized or found it obvious that UEs/terminals in TS-23.140 would include a "processor" capable of implementing Ellison's techniques.

292. Moreover, as noted above, it was well-known before the Critical Date for operating systems to include the messaging services and clients/agents responsible for provisions of those services, within the trusted computing environment, and for applications (including those that would be intended recipients of those messages) to be tiered/organized outside this environment.

293. Given that TS-23.140 contemplates mobile terminals with an MMS User Agent that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging) from other applications/components of the environment, and Ellison's own disclosures confirming the same and providing such tiered, security architecture within the device, a POSITA would have been motivated to similar enable such improved security within the MMS environment by implementing the MMS User Agent as part of the trusted computing environment (i.e., the secure execution environment) and providing the applications that communicate with this agent outside this trusted/secure environment.

294. For the above reasons, TS-23.140 renders obvious that the MMS User Agent executes in a secure execution environment and one or more of the applications to which push messages are routed executes outside of the secure execution environment.

D. Ground 1D: Claim 8 Is Rendered Obvious By TS-23.140 and Rakic

1. Claim 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.*

295. For the reasons described below, TS-23.140 in view of Rakic renders claim 8 obvious that message link server (e.g., the MMS Relay/Server) comprises a secure server (e.g., a secure push message server) to provide secure authorization signatures to the given one of the wireless end-user devices (e.g., UE/terminal), the secure authorization signatures indicating the authorized software components (e.g., applications receiving application data that have previously undergone an application registration process) that are to receive data from secure message link messages via the message link server.

296. In more detail, TS-23.140 discloses an "application registration process" that registers an application with the MMS User Agent and allows it to use MMS to transport application-specific data. Ex-1004, §7.1.18, p. 54.

297. As a result, when an application is registered and subsequently receives an application-specific message, that message is routed to the application based on the application identifier of the application as included in the message. *Id.*, pp. 54-56 ("If the destination application resides on a receiving MMS VAS Application, the MMS VAS Application shall immediately route the received MMS information on to the destination application that is referred to by the destination application

identifier ***(based on the negotiated details upon application registration process)"***; *id.* ("***If the MMS Relay/Server finds from the recipient MMS User Agent's capability indication (see clause 7.1.3.1) that the recipient MMS User Agent supports transport of application data***, the MMS Relay/Server shall not perform any type of content adaptation to a multimedia message (MM) that may be contained in the payload of an abstract message that contains a destination application identifier; … shall pass on the destination application identifier, the 'reply-path' identifier (if present) and the additional application/implementation specific control information (if present) unaltered").

298. Specifically, if the destination application resides on a receiving MMS User Agent and is registered, "the MMS User Agent shall immediately route the received MMS information on to the destination application that is referred to from the destination application identifier ***(based on the negotiated details upon application registration process)*** without presentation to the user." *Id.*

299. In this manner, the MMS User Agent includes data indicating the authorized/registered applications (i.e., software components) that are allowed to receive data from secure message link message via the MMS Server (i.e., message link server).

300. TS-23.140 additionally discloses that, if the destination application identifier is present in the message, the message is directly routed to the destination

application. *Id.* However, TS-23.140 does not expressly describe validation or authentication of the message that verifies the identity of the sender of the message.

301. Without such verification, a POSITA would have understood that the device and/or the destination application can be vulnerable to attacks by unknown third-party senders. Nor does TS-23.140 verify that the message is indeed intended for the correct application, which too can inadvertently result in confidential or sensitive data or information leaking to an incorrect device or an application.

302. MMS communications are a type of push communications. *Id.*, p. 10 ("MMS uses a number of technologies to realise [sic] the requirements of the stage 1 description (3G TS 22.140) [1]. The present document describes how the service requirements are realised with the selected technologies. As far as possible existing protocols (e.g. WAP, SMTP, ESMTP as transfer protocols; lower layers to provide push, pull, notification) and existing message formats (e.g. SMIL, MIME) shall be used for the realisation [sic] of the Multimedia Messaging Service.").

303. And, in the context of push communications, Rakic provides one well-known solution to address the above-described computer security concerns. As described in §VII.C, Rakic discloses a database device 108 that stores information related to client devices (e.g., codes obtained from a client device such as IMEI or a component in the client device (e.g., removable memory), and a secure push message server 504 uses stored information in the database device 108 to generate a key, an

electronic signature, and a secure push message for a push message intended for a particular client device. Ex-1015, [0047]-[0049], [0065].

304. The secure push message server generates the key based on the above-described code(s) (e.g., IMEI) that are obtained from client device 102 or a component thereof, and subsequently, generate an electronic signature based on the key and the message. *Id.*, [0066].

305. The secure push message server 504 generates and sends the secure push message to client device 102, e.g., by appending the electronic signature to the message and/or other information (e.g., a timestamp). *Id.*, [0067]. At the client device, a secure push message client can validate the message and the sender by generating a key and then generating a signature using this generated key and the data block of the received message. *Id.*, [0109]-[0112]. If the signature portion of the received message matches the signature portion of the received message, the secure push message is determined to be valid. *Id.*

306. A POSITA would have been motivated to implement the above-described techniques in Rakic within the MMS Relay/Server to address the above-described security problems. Indeed, a POSITA would have been motivated to do so because, in view of Rakic, the above-described techniques enable (1) the client device to "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." *Id.*, [0041].

307. In doing so, a POSITA would have recognized the improvements to data and computer security for both the client device and the underlying applications to which the data is being transmitted. *Id*. Similar modifications to email, in the form of the Secure/Multipurposed Internet Mail Extensions (S/MIME) (B. Ramsdell, *S/MIME Version 3 Message Specification*, IETF RFC 2633 (June 1999)— Ex-1035) and Pretty Good Privacy (PGP) (Philip Zimmermann, *Pretty Good Privacy: RSA Public Key Cryptography for the Masses,* PGP User's Guide. Version 1.0 (June 1991)—Ex-1034), were long known in the art as a means of providing cryptographic guarantees of confidentiality and authenticity in messaging.

308. In view of the above motivations, a POSITA would have implemented Rakic's secure push message server 504 and database device 108, as well as their above-described functionality, as part of the MMS Relay/Server in TS-23.140.

309. So, configuring the TS-23.140 system to implement Rakic's teachings would have amounted to the use of a known technique to improve similar devices in the same way, and combining prior art elements according to known methods to yield predictable results. Indeed, given that TS-23.140 and Rakic both contemplate push messaging with a similar push architectures/systems involving a push server (e.g., MMS Relay/Server and Secure Push Message Server) that communicates with

a client/user device, a POSITA would have found it straightforward to implement—with a reasonable expectation of success—the signature generation functionality of Rakic's secure push messaging server as well as its database storing the client device information, as part of the MMS Relay/Server in TS-23.140.

310. Additionally, the resulting system's elements would each perform functions they were known to perform prior to the combination—i.e., TS-23.140's system would continue to facilitate communication of messages between MMS Relay/Server and MMS User Agents, and Rakic's teachings (as implemented in combination) would facilitate provision of secure signatures (along with the messages) to the message receiving device(s).

311. In the resulting TS-23.140-Rakic combination, Rakic's secure push message server and its associated signature generation functionality would be implemented in the MMS Relay/Server and would be used to generate signatures for messages intended for different UE/terminals and their respective MMS User Agents.

312. A POSITA would have understood that the signatures are authorization signatures because they are used for authentication of the sender and/or verify that the intended recipient of the message is the client device and the client device includes a correct component. Ex-1015, [0041].

313. Additionally, as Rakic explains, the signature generating functionality "verif[ies] that an intended recipient of the secure push message is the client device, and that the client device includes a correct component." *See id.* A POSITA would have found it obvious that one such component could be an application of the device. In other words, a POSITA would have understood that Rakic does not limit its teachings of verification of component to hardware components, and in the context of push communications that are intended to be delivered to specific software components/applications, a POSITA would have found it obvious that such components could include software components.

314. As such, in view of Rakic as applied in the context of TS-23.140, a POSITA would have found it obvious that the above-described secure authorization signatures indicate the authorized application (i.e., software components) that are allowed/authorized to receive messages including application data via the MMS Relay/Server.

E. Ground 1E: Claim 9 Is Rendered Obvious By TS-23.140 and Adamczyk

1. ***Claim 9:*** *The message link server of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

315. TS-23.140-Adamczyk renders obvious a message delivery trigger being the expiration of a periodic timer.

316. Adamczyk discloses a message-transmission system like TS-23.140. *See, e.g.*, Ex-1021, Abstract, [0007]-[0009]. Adamczyk, however, expressly describes how notification servers (like TS-23.140's MMS Relay/Server) can be "configured to send [] notification messages to the recipient on demand, at a specific future time, and/or on a periodic schedule" including based on the preferences of the user. *Id.*, [0010], [0022], cl. 1.

317. Although TS-23.140 discloses a series of automatic and manual triggers (which I describe in (1[d3]/15[d2]) and incorporate here), in my opinion, a POSITA would have appreciated that TS-23.140's (automatic) triggers could be supplemented with the additional types of triggers like a periodic schedule expressly disclosed by Adamczyk.

318. In my opinion, a POSITA would have been motivated to add an additional periodic timer trigger to the list of triggers that TS-23.140 already discloses for several reasons. First, additional message delivery triggers create a more configurable system that better suits a wider range of user needs in an efficient manner. For instance, rather than a recipient MMS User Agent specify a single deferred time of delivery, or require the MMS User Agent to perform periodic polling to retrieve messages from external networks, a POSITA would have been motivated to implement a periodic timer like Adamczyk's to have the MMS

Relay/Server issue notifications at pre-determined intervals will all messages received during that interval.

319. Second, a POSITA would have appreciated that periodic timers would arguably improve the battery life of the device operating the MMS User Agent. Rather than have to establish a connection between the MMS User Agent and MMS Relay/Server every time a message is received by the MMS Relay/Server (causing a notification to be sent to the MMS User Agent when in automatic retrieval mode and no deferred delivery), or have the MMS User Agent repeatedly ping the MMS Relay/Server to see if messages are pending, the number and timing of connections and retrievals could be limited to pre-set intervals. *See, e.g.*, Ex-1040, [0074] (describing battery benefits of setting intervals for call connections rather than persistent connections).

320. In my opinion, a POSITA would have had a reasonable expectation of success in adding the periodic timer trigger of Adamczyk to the TS-23.140 system because the references suggest such a combination. As discussed above, Adamczyk expressly recognizes that its teachings can be used to augment existing 3GPP notification networks. Ex-1021, [0017]. Additionally, TS-23.140 already discloses a periodic polling method of retrieval in which a UE regularly requests retrieval from an MMS Relay/Server. Augmenting the Relay/Server to include periodic notification pushes based on some configurable timer would be another

implementation that would include the above noted benefits. Configuring the Relay/Server to push messages at predetermined, periodic intervals, would have merely required configuring the Relay/Server to hold received messages in the memory store/MMBox until expiry of a regularly recurring timer or periodic scheduled event before pushing all messages received during the time interval at once.

F. Ground 1F: Claim 10 Is Rendered Obvious By TS-23.140 and Herzog, or TS-23.140, Adamczyk and Herzog

1. ***Claim 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.*

321. TS-23.140-Herzog, or TS-23.140-Adamczyk-Herzog renders obvious setting the period of the timer fractionally shorter than a maximum data-message interval after which the secure connection is disabled.

322. As I described previously in two assessments of claim 9 (one in view of TS-23.140 alone, and another in view of TS-23.140 and Adamczyk, incorporated here), TS-23.140 describes a periodic timer through periodic polling, and TS-23.140-Adamczyk together teach a periodic schedule for message delivery. These references do not, however, appear to expressly describe what happens to the connections between the MMS User Agent and MMS Relay/Server to account for these periodic schedules.

323. Herzog is one example of prior art, however, that taught mechanisms to control the duration of a connection between a client (like TS-23.140's MMS User Agent) and server (like TS-23.140's MMS Relay/Server) through keep-alive timers to periodically ping a server to keep the connection alive, or to terminate the connection when the keep-alive messages are not sent for a given period of time. Ex-1008, [0036], [0041]-[0043].

324. In my opinion, a POSITA would have been motivated to implement keep-alive timers and messages in the TS-23.140 and/or TS-23.140-Adamcyzk system to control the duration and termination of a connection between the MMS User Agents and MMS Relay/Server automatically. Specifically, connections would be maintained for as long as the MMS User Agent indicates it is available and would automatically disconnect when the MMS User Agent stops sending keep-alive messages. Incorporating a keep-alive timer would have several benefits. First, the MMS Relay/Server would be able to track the availability and connection status of each MMS User Agent. Second, the MMS Relay/Server would avoid trying to maintain connections with disconnected/unavailable MMS User Agents. This would, in turn, reduce bandwidth demand and provide openings for other MMS User Agents to connect to the MMS Relay/Server to the extent the availability of a connection was a bottleneck (e.g., during periods of high demand).

325. In my opinion, a POSITA would have had a reasonable expectation of success in making this combination because TS-23.140 discloses a similar messaging system to Herzog, including a gateway intermediary between a server and a client. Ex-1008, [0005], [0024] (explaining that the server "is any type of server that is programmed to push content to the client"); TS-23-140, 30. And like TS-23.140, Herzog discloses that messages can be "saved in a message buffer…until [the message] can be delivered to the client" (e.g., when the connection is kept open by keep-alive messages). Ex-1008, [0031]. Herzog also discloses using various types of connections, including IP, TCP, SSL and HTTP. *Id.*, [0026], [00037]. As a result, the combination would only require implementing Herzog's keep-alive-messaging between the MMS User Agent and MMS Relay/Server to maintain TS-23.140's existing connection.

326. Moreover, upon making this combination, in my opinion, a POSITA would have understood that the period of the periodic polling (disclosed by TS-23.140) or periodic message delivery schedules (taught by TS-23.140 combined with Adamczyk) would be some degree shorter than the period required for the keep-alive timers to indicate availability of the MMS User Agent. If the periodic polling/message delivery were set to a longer period than the keep-alive timer period, the connection may be terminated before the periodic polling/delivery message could be sent, thus requiring a new connection to be established. And

because the ’192 Patent does not specify what constitutes “fractionally shorter,” in my opinion, any degree shorter can be considered “fractionally shorter,” as claimed, since it would be some “fraction’ of the duration of the keep-alive timers.

G. Ground 1G: Claim 14 Is Rendered Obvious by TS-23.140 and Gellens

1. ***Claim 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.*

327. TS-23.140-Gellens renders obvious a message delivery trigger based on an amount of wireless network data usage consumed by an end-user device.

328. As I explained previously in 1[c1]/15[c1] and 1[d3] (incorporated here), TS-23.140 discloses several triggers to control when messages are delivered to an MMS User Agent from authorized applications. EX-1004, 23, 30-31, 54-55. However, TS-23.140 does not appear to disclose triggering message delivery based on an amount of data consumed by the UE/mobile device.

329. As of the Critical Date (2009), however, wireless bandwidth was not as extensive as it is today and transmission costs per unit data were generally higher. Prior art, such as Gellens, recognized a need to control the delivery of data over cellular networks due to issues such as cost and bandwidth. Ex-1020, Abstract. Specifically, Gellens recognized that applications incorporated into network systems were sending an increased number of messages to end users—many of which were

"low priority," yet transmitting these messages could drive up costs to establish the connections to deliver these messages, leading to "excessive and wasteful traffic usage." *Id.*, [0004] ("Especially when the amount of data to be exchanged is small, the cost of bringing the traffic channel up and down becomes high compared to the amount of data exchanged. By contrast, when a larger amount of data is exchanged, the cost of bringing the traffic channel up and down is spread over more traffic."); *see also id.*, Abstract, [0002]-[0006], [0027], [0029].

330. To address this problem, Gellens proposes using triggers (like TS-23.140) to control when messages are delivered from applications to clients. Gellens provided this control by allowing applications to be assigned priorities, and messages from low-priority applications would be delivered only after a higher-priority application initiated a link and transmitted its message. *Id.*, [0035]-[0036]. That way, connections would not be established and maintained for *every* message—no matter how minor—but low-priority messages would be queued until a message of significance opened the connection. In other words, the "trigger' would be the high-priority application using "an amount of wireless network data," as claimed, which would then trigger delivery of the low-priority messages.

331. In my opinion, a POSITA would have been motivated to combine TS-23.140 and Gellens to prioritize certain applications, and only permit transmission of low-priority application messages after a high-priority application opens a

connection and transmits its message, thereby limiting connection/transmission costs, and "the cost of bringing the traffic channel up and down is spread over more traffic." *Id.*, [0004], [0035]-[0036]. In my opinion, this combination would have amounted to applying a known technique (application prioritization) to a prior-art device (an MMS network) to solve a known problem in the art (networking costs) and achieve a predictable result (triggering low-priority message delivery only after a high-priority application establishes a connection).

332. In my opinion, a POSITA would have also had a reasonable 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 MMS Relay/Server and MMS User Agent.

H. Ground 2A: Claims 1, 5-7, 9-13, and 15 are Rendered Obvious by Houghton and Munson

1. Combination of Houghton and Munson

333. In my opinion, a POSITA would have found it obvious to combine the teachings of Houghton (Ex-1007) and Munson (Ex-1017), and specifically to incorporate Munson's store-and-forward functionality into Houghton's push messaging system.

334. Specifically, in my opinion, a POSITA would have been motivated to incorporate computer memory and programming to deliver messages from

memory—consistent with Munson's teachings—to improve flexibility and reliability of the system.

335. For instance, a POSITA would have recognized or found it obvious that a mobile terminal that is an intended recipient of a message may not have network connectivity at the time a message originator initiates delivery of a message. In this instance, the message would be stored in the computer memory until the mobile terminal reconnects to the network and the message can be delivered. Indeed, Houghton already contemplates such "store and forward messaging systems," noting that messages may not be deliverable while the user is out of network coverage. Ex-1007, 3, 7 ("These server computers such as a telecommunications operator SIP Server or WAP Gateway is used to control and deliver messages and ***store messages when the mobile terminal is unavailable and deliver them on the sender[']s behalf, possibly later then the terminal becomes available***"). Moreover, it was well-known for store-and-forward push systems to include buffers to store messages "when the mobile terminal is not connected to the mobile internet service network" and "re-transmitting the stored push information when the mobile terminal is connected to the mobile internet service network." Ex-1006, [0034], claim 14, Fig. 1. A POSITA would have therefore been motivated to implement computer memory and programming to deliver messages from memory, consistent with Munson's message-buffering solution to avoid losing or discarding of messages (e.g., when the

mobile terminal is outside network range or connectivity and the message is therefore undeliverable).

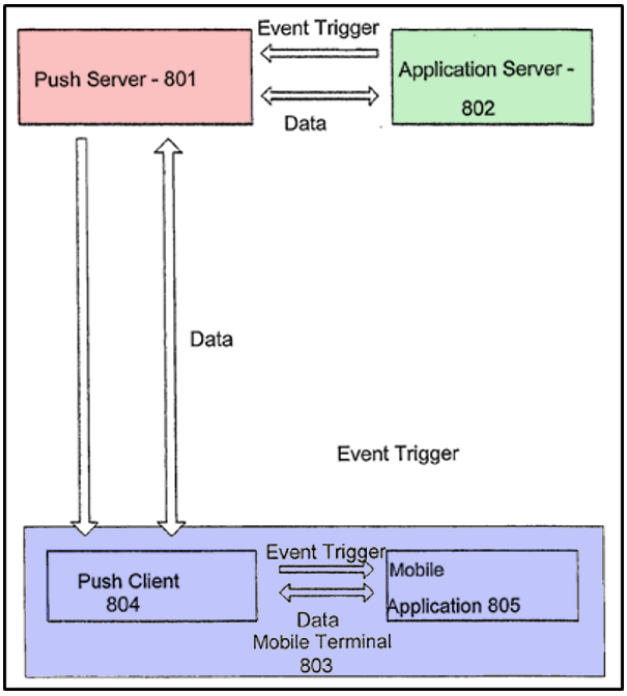
336. Moreover, a POSITA would have appreciated that, by providing computer memory to store messages mid-transmission, both the originator and recipient would be able to better control the conditions of message delivery. Houghton, for instance, discloses that the "push message from the push server 401 may be triggered by any trigger event, local or remote, ***defined at the server***. The trigger may, among others, include an alarm, notification, or measurement result received to the push server 401 from another device or system." Ex-1007, 21; *see id.*, 14 (describing various trigger events), 21-22, 25-28 (describing various trigger events). Munson similarly teaches messages can be delivered from memory "according to a schedule of time ***or event***" such as coordinating pushes "for a system maintenance purpose during off-peak hours." Ex-1017, [0040], [0044]. By providing memory along the transmission path to buffer messages, both the originator and recipient would be able to define additional conditions to complete delivery of the message without having to rely on, e.g., the availability/connectivity of the originator/recipient devices once a delivery condition is triggered; they would be delivered from the intermediate memory instead. *See also, e.g.*, Ex-1004, 27-29 (describing both originator-defined and recipient-defined delivery conditions to deliver messages from a central memory buffer).

337. In my opinion, combining Houghton and Munson's teachings would have been obvious and straightforward because such a combination would have amounted to using known prior art techniques (message buffering) to improve similar devices/systems (push messaging systems, as contemplated by Houghton and Munson) in the same way that yields predictable results (push messaging systems that can store and forward messages and subsequently deliver them, as contemplated by Houghton and as expressly described by Munson).

338. A POSITA would have had a reasonable expectation of success in making this modification because Houghton already envisions not only store-and-forward systems that provide centralized computer memory to buffer messages, but also conditioning delivery upon the occurrence of certain events. Ex-1007, 3, 7, 14, 21-22, 25-28. All Munson provides is additional details about how to implement a store-and-forward system by providing the computer memory at a server, and the computer programming to deliver messages from memory (including upon occurrence of certain events). Ex-1017, [0036]-[0038], [0044], Fig. 4. The resulting combination would merely require generic computer memory (at a central server) to serve as the message buffer, and programming to store incoming messages in memory and then deliver them from memory (including at the occurrence of a triggering event both Houghton and Munson already envision).

339. Thus, in the combination, each of Houghton and Munson's systems would continue to operate the way they did prior to combination, e.g., Houghton's system would continue to push messages between a push server and a push client, and Munson's described techniques (as implemented in combination) would facilitate buffering of push messages in a buffer/memory of Houghton's push server and delivering the stored messages when certain conditions arise (e.g., user becomes available in network range).

340. In an example of the resulting combination (referred to as Houghton-Munson) and as illustrated below, Munson's handler units and buffer (which handle and store messages to ensure delivery) and content senders (which coordinate message transmissions) would have been incorporated into Houghton's push server such that they would buffer push messages based on the policies (as further described by Munson) and subsequently deliver those messages upon occurrence of certain conditions. *Id.*, 3, 7, 11-14, 21-23, Figs. 3-4, 8; Ex-1017, [0036]-[0038], Fig. 3-4.



Ex-1007, Fig. 8 (modified to incorporate Munson).

341. Finally, in my opinion, a POSITA would have been further motivated and had a reasonable expectation of success in implementing some of the additional delivery conditions Munson taught in Houghton's system—to the extent Houghton does not already disclose the same triggers as Munson—to provide additional flexibility and conditions to control delivery. Houghton, for instance, discloses that message delivery may be conditioned upon "***any trigger event***, local or remote," Ex-1007, 21, and therefore already contemplates additional conditions may be incorporated into the system. All Munson provides are some additional or alternative "trigger events." Ex-1017, [0040], [0044]. Implementing additional trigger events in Houghton's system would have required only some computer programming at the

server, which, in my opinion, would have been well within the skills of a POSITA to implement.

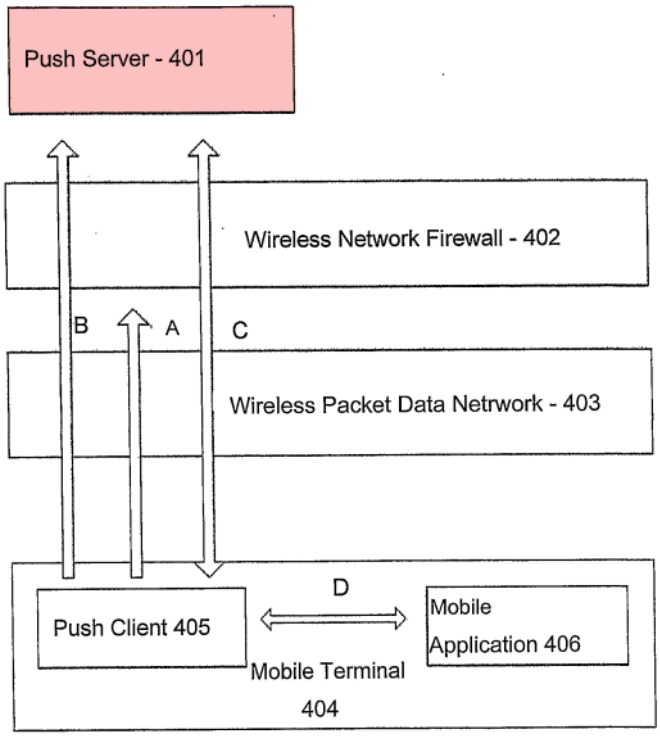
2. Claim Analysis

a. Claims 1 and 15

*i. **1[pre]:** A message link server comprising / **15[pre]:** A method of operating a message link server, comprising:*

342. If the preamble is limiting, Houghton-Munson renders obvious a message link server (e.g., push server 401)[6] and a method for operating the same. As described below, and shown with reference to Fig. 4, Houghton's push server 401 sends push messages over communication links formed over a network to push client (405/804) on mobile terminals (404/803) for delivery to mobile applications (406/805).

[6] Houghton provides several figures and/or embodiments that use different reference numbers for the same structures (e.g., "push server" is provided reference number 401, 501, 701, 801, 901). Although Houghton describes these as different embodiments, a POSITA would have understood that, for the most part, their teachings are not mutually exclusive of one another, but instead describe different aspects or procedures, sometimes involving additional network elements, that a single system may be configured to carry out. *Compare, e.g.*, Ex-1007, 15-16 (noting Figure 4 describes one "mobile terminal" embodiment of the network system, Figure 6 depicts a process to maintain an open network path across the network, and Figures 7-9 depict different message-push procedures that incorporate an additional "application server" into the network), *with id.* (noting Figure 5 describes an *alternative* wired terminal embodiment). For this reason, I often cite to reference numbers from Figure 4, but I also rely upon other teachings provided in the context of other figures (e.g., Figures 6-8), because the features associated with these other "embodiments" may obviously be implemented together.



Ex-1007, 15, Fig. 4 ("illustrating an exemplary implementation of the system for pushing mobile application launch commands to a mobile terminal based on server-side events in accordance with one embodiment of the present invention").

343. "Figure 4 illustrates a Push Client basic functionality 405 implemented in a mobile terminal 404 ***as part of a message passing system in cooperation with a Push Server 401***. The push client 405 is preferably implemented in the form of a software run in a processor of the mobile terminal 404. … The push client software 405 runs in the background and attempts to log into the push server 401 over a data network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets." Ex-1007, 16-17; *see id.*, Abstract ("A server outside a wireless

network security entity is adapted to push messages to a mobile terminal located in the wireless network and thereby cause programs to start to a specified operating state or, if already started, change operating state to the state indicated by the message thereby presenting a new application view to the user of the mobile terminal. ***This mobile-initiated permanent connection allows the server to pass or push messages from outside the operator's firewall to the mobile terminal at any time***."); Figs. 4, 8.

344. Additional details regarding Houghton-Munson's push server and the method for operating this server are described with reference to 1[a]-1[d4]/15[a]-15[d3] (which I incorporate here).

345. For the above reasons, Houghton-Munson renders obvious 1[pre]/15[pre].

ii. ***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, /* ***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,*

346. Houghton-Munson renders this limitation obvious consistent with how this limitation is described in the context of the '192 Patent's specification. Specifically, the '192 Patent explains that the "transport services stack" facilitates

communications between the server (service controller 120) and the client device (service processor 115) over a communication link (service control link 1653). Ex-1001 (’192 Patent), 90:33-38.

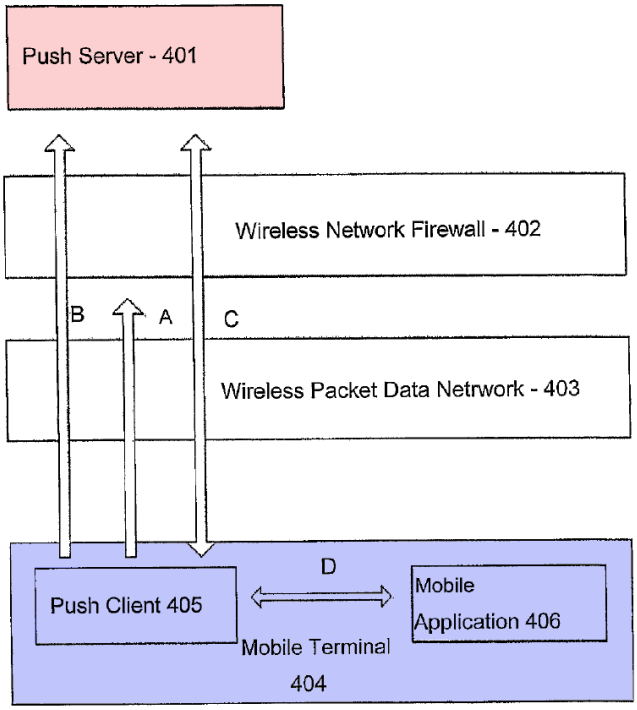
347. The patent further discloses use of transport protocols for such server-client communications, including, e.g., TCP, IP, UDP (*id.*, 99:8-32) and notes that “there are many other approaches for implementing a secure control channel over the Internet including, for example, one or more of various packet encryption protocols running at or just below the application layer, running TCP Transport Layer Security (TLS), and running IP level security or secure tunnels” (*id.*, 17:13-22).

348. Consistent with these disclosures, Houghton-Munson renders obvious a transport services stack that maintains a secure message link through an Internet network between the message link server (e.g., push server 401) and a device link agent (e.g., push client 405) on respective client devices (e.g., mobile terminals 404). Houghton discloses that 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”). Ex-1007, 17, Fig. 4 (below; annotated).

349. For example, and as explained below, Houghton discloses communications over the Internet network between the push server 401 and push

client 405, using various transport protocols that have additional security to facilitate secure links between the push server 401 and the push client 405.

350. Specifically, Houghton's push client software 405 on mobile terminal 404 "runs in the background and attempts to log into the push server 401 over a data network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets. The arrow A in Figure 4 illustrates an optional attempt by the push client 405 to communicate with the wireless network firewall 402 or other similar security or gateway device in order to open a connection for the push server 401 to directly contact the push client 40[5] as in FIG 3. If successful, operations can continue as described in FIG 3 or as below. Arrow B illustrates a login attempt." *Id.*, 17, Fig. 4 (annotated below; showing communication between push client 405 and push server 401 over a wireless network).



Id., Fig. 4.

351. Push client 405 constitutes the device link agent because it is "software" running on the mobile terminal device that "initiates and maintains" a link using "Internet technologies" between the mobile terminal 404 and the push server 401, and using this link, push server 401 pushes messages to the mobile terminal. *See, e.g.*, Ex-1007, 11, Abstract ("A server outside a wireless network security entity is adapted to push messages to a mobile terminal located in the wireless network and thereby cause programs to start to a specified operating state or, if already started, change operating state to the state indicated by the message thereby presenting a new application view to the user of the mobile terminal. ***The mobile terminal is provided with means, such as software, which initiates and maintains contact with the***

server using Internet technologies. This mobile-initiated permanent connection allows the server to pass or push messages from outside the operator's firewall to the mobile terminal at any time."), 19-20 (describing a data structure stored in mobile terminal including preferred communication protocol and other network connection parameters for facilitating connection between push client and push server), 20 ("At any time after a connection is established between the client 405 and the server 401, the client may push a message to the server or the server may send a push message to the client. This is illustrated by means of the arrow C in Figure 4. ***These messages through a previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS arrive quite rapidly***…. ***A connectionless protocol, such as UDP/IP, may also [be] used in cases where the connection-oriented protocol is not possible or not desirable because of the push server resources used by every static instance of a connection-oriented protocol***. Connectionless communication is also used when a message needs to get through to the server 401 quickly, such as when occurs when the client 405 notifies the server 401 of a[] terminal event, user interface event or application event but the connection-oriented protocol connection is not connected or is already busy transferring information…."); Ex-1001 ('192 Patent), 43:18-50 (interchangeably using agent and software (such as an application) running on a device).

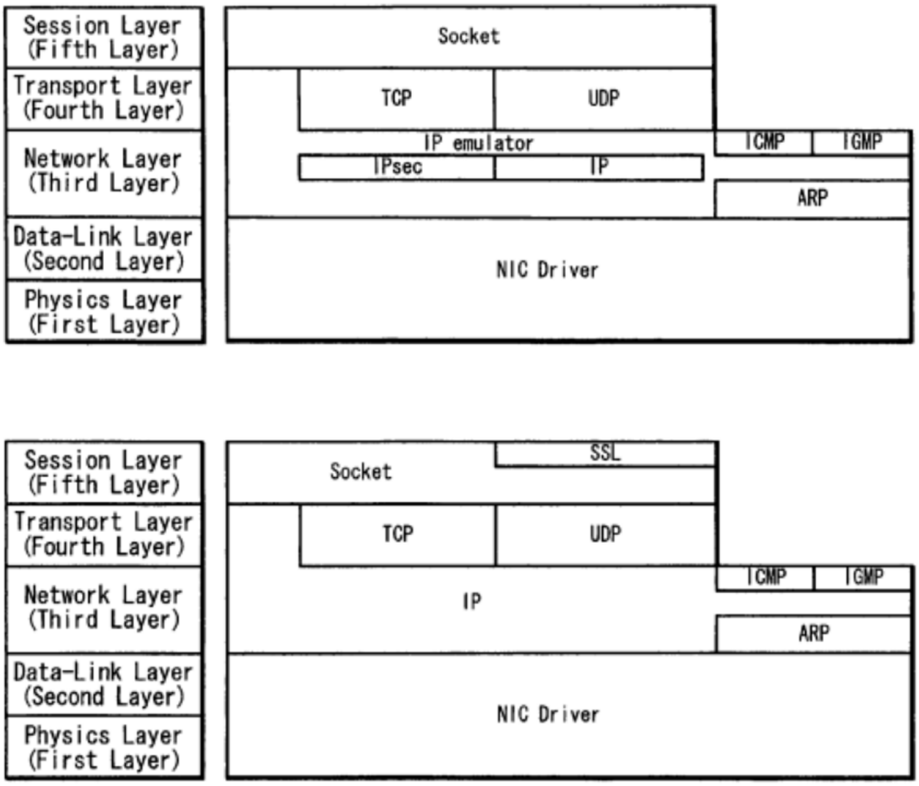
352. For this connection, Houghton contemplates use of known transport protocols, including, e.g., "a connection-oriented protocol suitable to the wireless network such as TCP/IP," "a connectionless protocol suitable to the wireless network 403, such as UDP/IP," "an alternate connection-oriented protocol such as HTTP suitable for mobile push operation through more strict firewalls and proxy Internet connections," or "a secure protocol such as HTTPS, IP-Sec, secure IP6 or a proprietary security protocol to identify the communicating parties, prevent message interception by 3rd parties, and prevent message modification by 3rd parties." Ex-1007, 18-20; *see id.*, claim 10 ("method as claimed in claim 9, wherein said trying of one or more other techniques and/network parameters include one or more of using an alternate push server network address which may be more available or more accessible at the present time and from the present location, using one or more alternative mobile terminal network connection settings, or variation of the parameters within the settings, using an alternate connection route through a second wireless or fixed terminal accessible to the wireless terminal over a wireless network and capable of communication with the push server, using a TCP connection, using a UDP connection, using an SSL connection, using an HTTP or tunneling HTTP connection, using an SHTTP or tunneling HTTPS connection, using an XML-based connection over other transport protocol.").

353. "At any time after a connection is established between the client 405 and the server 401, the client may push a message to the server or the server may send a push message to the client," which is illustrated using arrow C in Figure 4. *Id.*, 20 (explaining using "a previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS" for communication between push client and push server).

354. The '192 Patent does not explicitly describe the structure of the "transport services stack" and instead only describes it functionally in terms of it facilitating transport/communication of data packets over an Internet network and between a server and a client. In some embodiments, the '192 patent describes the communication as being secured using encryption, TLS/SSL, HTTPS, IP-Sec, or another security mechanism. Ex-1001 ('192 Patent), 16:66-17:22, 39:20-32, 90:33-61, 99:8-32; 100:8-32. The above descriptions in Houghton similarly describe the server-client communications using such secure transport protocols. From these disclosures, a POSITA would have understood or found it obvious that Houghton's push server would include a "stack" of transport protocols to facilitate its secure communications over an Internet network with the mobile terminal 404. Ex-1007, 1 (noting TCP and UDP protocols—which operate at the "transport" layer of the OSI model—"run on top" of "IP protocols") 17-20, Fig. 1.

355. Indeed, before the Critical Date, it was well-known for a device implementing transport protocols (as already contemplated by Houghton) to implement a transport services/protocol stack that facilitates secure network communications with other network elements (e.g., a mobile terminal).

356. For example, a prior art reference, Ozaki, describes the state of the art for TCP/IP protocol stacks that are implemented in a device's "software or firmware" and that are made up of multiple layers, including a physical layer, a data link layer, a network layer, a transport layer, a session layer, a presentation layer, and an application layer. Ex-1005, [0012]-[0014]. The physical and data link layers are typically implemented using a device's network interface card (NIC) and an associated driver, and enable connecting the device to the network. *Id.*, [0015]-[0018]. The network layer "defines a communication method between two … nodes" and in the "case of TCP/IP" this layer corresponds to an IP layer, and the transport layer corresponds to the TCP layer. *Id.*, [0019]-[0021]. IPsec (Internet Protocol Security) is used to encrypt communications in the network layer. *Id.*, [0019]-[0021], [0008]-[0010] (describing use of symmetric/common key encryption for IPsec encryption). The session layer "defines a procedure of a session (from start to end of a communication) and enabling a connection between applications," and Secure Socket Layer (SSL) is "conventional representative encryption communication technology" for facilitating encryption of communication in this layer. *Id.*, [0022].



Ex-1005, Figs. 25-26.

357. Given that Houghton already contemplates a stack of TCP/IP-based transport protocols for facilitating transport of data/messages between network entities, a POSITA would have understood that Houghton would have implemented any well-known TCP/IP transport stacks in the software and hardware of the push server to enable such data transport over a network.

358. Moreover, as described above, a communication link is established between the push client/mobile terminal and the push server 401, and this link constitutes a secure message link because it facilitates sending of push messages between the push client 405 and push server 401 using secure/encryption protocols

such as IPsec, HTTPS, and/or SSL. Ex-1007, 11, 18-20, claim 10; Ex-1001 ('192 Patent), 16:66-17:22, 39:20-32, 70:31-44, 90:33-61, 100:8-32, (similarly describing use of TLS/SSL, IPSec to secure network communications between network entities).

359. Given these disclosures in Houghton about using secure protocols such as SSL, HTTPS, and IPSec, and the well-known use of such protocols to secure TCP/IP-based network communications, a POSITA would have understood or found it obvious that the transport services stack that would be implemented in Houghton's push server would use such protocols for securing the TCP/IP based communication link between the push client 405 and the push server 401.

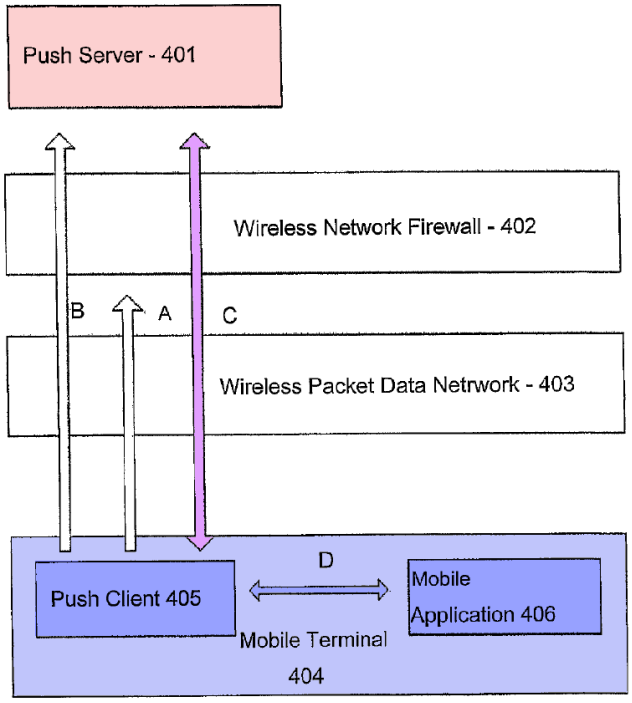
360. A POSITA would have further understood that, in push messaging environments such as those provided in Houghton, multiple mobile terminals (and their associated push clients) would be in communication with push server 401, such that a separate secure SSL or IPSec-based TCP/IP communication link would be between the push server and each mobile terminal and its respective push client. *See also* Ex-1007, 18, 19 (disclosing multiple "terminals" in Houghton's environment).

361. For the above reasons, Houghton-Munson renders obvious 1[a]/15[a].

iii. ***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; /* ***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;*

362. Houghton-Munson renders obvious that each of the wireless end-user devices (e.g., mobile terminals 404) include multiple software components (e.g., mobile applications 406) authorized to receive and process data (e.g., application-specific data intended for the application identified by the identifier and assuming user agent supports transport of application-specific data) from secure message link messages received via a device link agent on that device.

363. As shown below and described further in Houghton, wireless end-user device (mobile terminal 404) includes multiple software components (e.g., mobile application(s) 406), and messages (e.g., command push or application command messages) are provided to the respective applications via a device link agent (e.g., push client 405) on the device/mobile terminal 404. Ex-1007, 21-22 (disclosing messaging that identifies "which mobile application 406 from a plurality of such applications…. [T]he push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406"), 24-25, 27, Figs. 4, 8.



Ex-1007, Fig. 4

364. Push client 405 is "implemented in the form of a software run in a processor of the mobile terminal 404" and operates "as part of a message passing system in cooperation with a Push Server 401." *Id.*, 16-17; *see id.* ("The push client software 405 preferably launches automatically when the mobile terminal 404 is turned on. The push client software 405 runs in the background and attempts to log into the push server 401 over a data network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets. The arrow A in Figure 4 illustrates an optional attempt by the push client 405 to communicate with the wireless network firewall 402 or other similar security or gateway device in order to open a connection for the push server 401 to directly contact the push client 40[5] as

in Figure 3. If successful, operations can continue as described in Figure 3 or as below. Arrow B illustrates a login attempt."); *see also* Ex-1001 ('192 Patent), 43:18-31, 11:51-57 (describing the device agent as software executing in a device processor).

365. These messages are received over a secure TCP/IP link by push client (405) operating on mobile terminal (404) before being passed to the relevant application (406). Once a connection is established between push client 405 and push server 401, "the ***server may send a push message"*** to push client 405, as illustrated using arrow C above. Ex-1007, 20. Such a push message is transmitted to the push client (for delivery to a mobile application) through the "previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS." *Id*.; *see id.*, 34, claim 33 ("wherein ***the push client*** and push server establish a secure connection and ***unwrap[s] such security*** on behalf of the mobile application"); 16-17 (noting that the push client (405) operates in software); Ex-1001 ('192 Patent), 43:18-31 (explaining that device-link agents can be "implemented largely or entirely in software"). Because the communication link uses SSL to secure communications transmitted over the communication link between push server 401 and mobile terminal 404, the received push message from push server constitutes a secure message link message.

366. Additionally, a POSITA would have understood or found it obvious that push server 401 may communicate and facilitate transmission of push messages to multiple mobile terminals, each including its respective push client 405 and mobile applications 406. This is confirmed and corroborated by the following disclosure in Houghton: "The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 and direct such to the push server 401, possibly for redirection through push server API to other servers, ***mobile terminals, push clients and computing devices***." Ex-1007, 21. Therefore, the mobile applications 406 are configured to receive/process message data. *Id*.

367. The received push message include "application-specific data," such as "application commands," "updates," or "data messages" that are "***directed to a push application or dynamically loaded module or user interface feature from among a plurality of such applications, modules and features by means of said client-initiated or client-opened and client-maintained permanent connection***." *Id.*, claim 1.

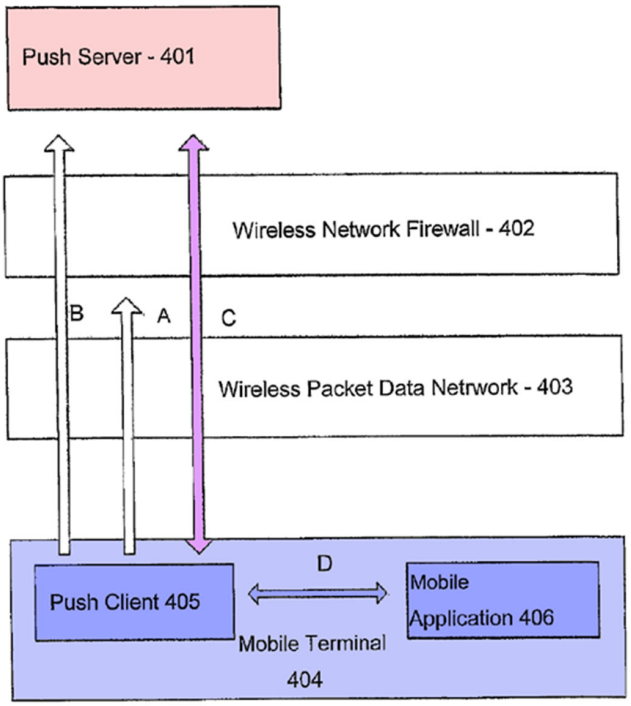
368. The received data is then processed by the receiving application to, e.g., present certain data or operate the application in a particular state. For example, "the push messages may include, for example, a message stimulating, triggering or ***commanding the launch, display or shift to foreground of a user interface feature of the terminal such as an application*** which is part of the basic terminal software

or and an additional installed application beyond the base software supplied with the mobile terminal." Ex-1007, 11.

369. In response, the "mobile terminal may spontaneously open the application, dynamically loaded module or user interface feature of the terminal thereof to a designed application view or location, present new information of the availability of new information, a new feature or operating state." *Id.*, 12. And, if already operating, "the action of such a message would instead stimulate the application or user interface feature to present itself to the foreground of the user interface and change state as specified in the application-specific message received from the server." *Id.*

370. The push message also includes "a data packet" with "binary or text information ***specifying which mobile application 406 from a plurality of such applications***" and additional "binary or text information to be passed to the … specified mobile application." *Id.*, 21, 22 ("The potential applications of the COMMAND PUSH include but are not limited to mobile applications for enterprise connectivity, back office integration, database access, scientific and industrial process and equipment monitoring and control, pictures transfer, mobile gaming, application data transfer, mobile electronic commerce, application alarms, reminders, multimedia interactivity, corporate calendaring, personal messaging and group messaging, etc.").

371. The push message is subsequently directed from push client 405 to the specified mobile application 406 via the communication path shown using arrow D in Figure 4 below. The figure also shows the communication relationship between the push server (401) and push client (405) using arrows B and C.



Ex-1007, Fig. 4, 21, 24.

372. A POSITA would have understood or found it obvious that push server (401) does not establish a link with just one mobile terminal (404) with one push client (405), but instead communicates push messages to multiple "mobile terminals," each including its respective push client (405) and mobile applications (406). Ex-1007, 21 (routing messages to "servers, mobile terminals, push clients and computing devices").

373. Moreover, because the push command messages are specifically routed to a particular application from among a plurality of applications running on the terminal, and the destination application then operates on the received message data (e.g., launch application, present content, etc.), a POSITA would have understood or found it obvious that the applications executing on the mobile terminal are authorized to receive and process the data included in the command push/application command messages.

374. Additionally, or alternatively, even if it is determined that Houghton does not explicitly disclose that the multiple software components (e.g., mobile applications 406) are ***authorized*** to receive and process data from secure message link messages, a POSITA would have found that to be obvious in view of the relevant state of the art, as explained below.

375. Before the Critical Date, it was well-known for applications that are intended recipients of push messages to register with a push server and/or push client before they can receive messages transmitted over such push frameworks. For example, in the relevant context of push communications and push environments, Lee discloses that "various push services provided to ***each of [the] applications are managed*** in an integrated fashion through the integrated push service server 100 and the integrated push service agent 250" and for this reason, specification of a recipient application (similar to Houghton's disclosures) are required in push messages to

facilitate appropriate transmission of the push message to the application. Ex-1006, [0023].

376. Similarly, Shenfield discloses another push environment in which push client includes "an application registration application provider interface adapted to register applications to said push client and further adapted to receive an application manifest for said applications." Ex-1018, [0017] (explaining that the push client may "register with only one application"), [0063], [0106]-[0109] (application "registers with push client 510, providing application manifest 918 to establish a channel to a content provider for servicing" the application); Ex-1004, 54-56 (explaining that "[a]pplications that want to transport data specific to applications other than MMS will initially need to register with the appropriate MMS User Agent or MMS VAS Application").

377. A POSITA would have found it obvious to implement the above-described and well-known authorization or registration processes in Houghton's system to enable registration of a particular application with a push client and the push server with which the push client interacts.

378. A POSITA would have been motivated to do so for multiple reasons, including to ensure the application is compatible with the push client/server's communication protocols and to enable dynamic content delivery "to have

information or data pushed" to devices without users of such devices having to "go and seek out that data." *See, e.g.*, Ex-1018, [0003]-[0006].

379. A POSITA would have been further motivated to implement application registration in Houghton's system to ensure that data and messages are routed to the correct application and simultaneously ensure that applications have opted into receiving push data/messages (thus avoiding inundating unregistered applications with content that they have not requested or opted into receiving). As an example, different applications likely expect incoming data in a variety of formats. Whereas weather data may come in one format (a series of numbers to be parsed and organized by the application), news applications are likely to send full text sentences to be read. Neither format in this example would make sense to the other application, and therefore delivering messages to them would result in confusion. As such, it is clear that a POSITA would have been motivated to ensure that information was routed to the correct application.

380. Moreover, because Houghton already contemplates that push client and push server coordinate on message delivery to and from applications resident on mobile terminals, implementing the well-known application registration/management teachings in push environments, within Houghton's similar push environment, would have been straightforward and a POSITA would

have had a reasonable expectation of success in combining the teachings in the above-described manner.

381. Therefore, a POSITA would have found it obvious that the push messages in Houghton are routed by push client 405 to applications that are authorized/registered to receive such messages.

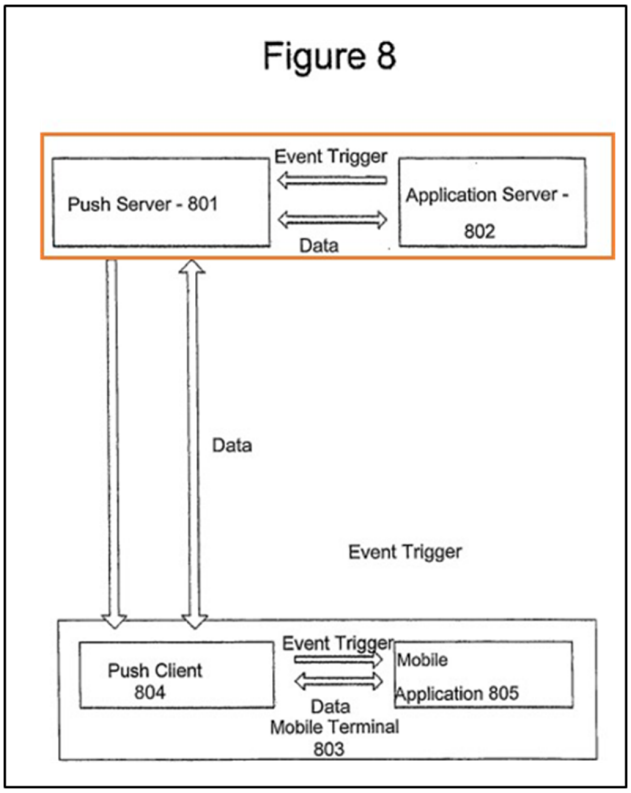
382. For the above reasons, Houghton-Munson renders obvious 1[b]/15[b].

iv. ***1[c1]:*** *an interface to a network to receive network element messages from a plurality of network elements, /* ***15[c1]:*** *receiving network element messages from a plurality of network elements,*

383. Houghton-Munson renders obvious an interface to a network to receive network element messages (e.g., an application programming interface (API)) from a plurality of network elements (e.g., application server(s) 702/802).

384. In more detail, Houghton's Figure 8 (reproduced below) illustrates an example of a "COMMAND PUSH" procedure where the 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 in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705***." Ex-1007, 21-22.

385. Fig. 8 of Houghton is shown below (annotated):



386. Thereafter, "a data connection between application server 802 and mobile application 805 is established with the aid of the persistent managed, tested and configured data connection in Figure 4, arrow C between push server 401/801 and push client 405/804" and "**[a]*n IP or other API (application programming interface) connection between application server 802 and push server 801 simplifies the work and cost of creating application server 802 by taking care of network details already established for the COMMAND PUSH procedure." Id.*, 22-23.**

387. Houghton further discloses that, "[i]n some cases a separate server such as a telecommunication service server, gaming environment server, and original

equipment manufacturer device or business information system may notify the server directly of events or messages to be sent using ***an external application programming interface (API)***. Such a device ***connected by external API*** may also receive information including the results and responses of messages which the user has replied to by terminal software or web page interaction." Ex-1007, 14; *see id.*, 21 ("The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 ***and direct such to the push server 401 , possibly for redirection through push server API*** to other servers, mobile terminals, push clients and computing devices.").

388. Moreover, the application server communicates over a network using the API or IP connection with push server 401. *See id.*, 23 ("***An IP or other API (application programming interface) connection between application server 802 and push server 801*** simplifies the work and cost of creating application server 802 ***by taking care of network*** details already established for the COMMAND PUSH procedure").

389. Indeed, before the Critical Date, it was well known for application servers and push servers to be distributed across a network, and for such servers to communicate over a network using an interface such as an API).

390. Additionally, a POSITA would have understood or found it obvious that the push environment contemplated by Houghton would include multiple

different application servers, each separately communicating with the push server to exchange data/messages with one or more mobile terminals that are connected to push server 401.

391. This is further corroborated by Houghton's following disclosure: "The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 and direct such to the push server 401, possibly ***for redirection through push server API to other servers***, mobile terminals, push clients and ***computing devices***." Ex-1007, 21.

392. Therefore, a POSITA would have understood or found it obvious that Houghton's push server includes an interface to a network in the form of one or more APIs via which application-specific messages and data are received, from one or more network elements such as application servers. Moreover, a POSITA would have understood that application servers are network elements and hence messages received from such servers would constitute network element messages.

393. For the above reasons, Houghton-Munson renders obvious 1[c1]/15[c1].

v. ***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; /* ***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;*

394. Houghton-Munson renders obvious that the received network element messages comprise respective message content, including data for (e.g., application-specific data and/or data related to the state in which to launch the server) and identification of a respective one of the authorized software (e.g., a specification of the application from among multiple applications to which to direct messages), and requests for delivery of the respective message content to respective wireless end-user devices (mobile terminals). 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 a mobile terminal (404/803). Ex-1007, 21-22.

395. In more detail, in Houghton, "the push server 701 is triggered by a trigger event as for example from an application server 702 to push an application

command message to the push client 704 and thereby initiate a mobile terminal client trigger event in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705." Ex-1007, 21-22.

396. Upon occurrence of this trigger, "a data connection between application server 802 and mobile application 805 is established with the aid of the persistent managed, tested and configured data connection in Figure 4, arrow C between push server 401/801 and push client 405/804." *Id.*, 23.

397. Houghton further explains that the message sent from an application server to an application on the user's terminal through push server 401/801 and push client 405/805 includes "a data packet" along with "[a]dditional binary or text information ***specifying which mobile application 406 from a plurality of such applications may optionally be*** included" as well as "***binary or text information to be passed to the default or specified mobile application***" **and** ***"additional binary or text information specifying how upon receiving such a message the client 405*** will notify and optionally ask the user if the action should be performed may optionally be included." *Id.*, 21; *see id.*, claim 1 ("***pushing commands and data to a wireless terminal*** from a push server" to a push client"), 8 ("When a remote program such as a push server 202 sitting inside the operator's firewall 201 and with permission to contact ports on the mobile terminal 203 sends an IP packet to the terminal (A), ***the IP terminal software 204 routes the packet with [its] commands or application-***

specific data to the mobile application 205 (B), launching the application if it is not already running."), 21 ("Upon receiving such a message***, appropriate automated actions and user interface media capabilities of the mobile terminal as specified in the command C from the server 401 are executed by the push client 405. The most common case is playing a sound, asking the user if the mobile application should be launched, and if affirmative passing the command details (arrow D) to the specified mobile application,*** module or additional feature within the same application suite, launching the application, module or feature if needed. Such action may include optionally proceed without audio, with network downloaded audio, or without first requesting confirmation from the user as specified by user preferences and command details."). There could also be a "data packet containing no information." *Id.*, 21.

398. In one particular example, Houghton discloses a "trigger event" or "application command message" issued by the application server, which triggers the push client to send an application message that 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" in the message. Houghton, 21-22.

399. By stating that the message "specif[ies] which mobile application 406 from a plurality of such application" to which the message is directed, (*id.*, 21) and that the message is directed to a particular "mobile application 705 from a plurality

of such applications" (*id*., 22), in my opinion, a POSITA would have understood that Houghton's messages include an application identifier.

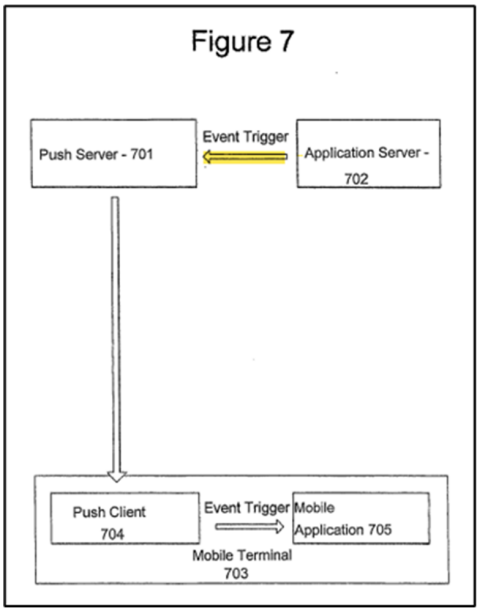
400. But even if Houghton did not expressly include an application identifier, in my opinion, including one would have been obvious. Before the Critical Date, it was well-known for such information specifying/identifying a particular application to be in the form an application identifier. Ex-1006, [0022]. For example, in the relevant context of push communications and associated environments, Lee discloses that the push message from a push server includes an "app_ID" that specifies a target application on the mobile terminal that is the destination for the message. *Id.*, [0011], [0013], [0022], claim 1. Similarly, TS-23.140 also contemplates that each destination application to which the push/MMS message is routed is identified using an application identifier in the message. Ex-1004, 54-56.

401. Given that Houghton routes messages to an application based on the "information specifying … [that] application" and given the common and well-known use of application identifiers to facilitate such identification, a POSITA would have found it obvious to implement Houghton's teaching of the "information specifying" the mobile application as the application identifier for that application. The messages received by the push server from the application server would include an application identifier because the application server is better equipped, relative to

the push server, to provide the application identifier since the application server originates the message and knows the intended application.

402. Implementing Houghton's system in this manner therefore would have amounted to an implementation of 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, as suggested by Houghton and as was well-known in the art before the Critical Date).

403. Moreover, a POSITA would have found it obvious that the received push message requests delivery of the message content to one or more mobile terminals. Specifically, and as explained above, "push server 701 is triggered by a trigger event as for example from an application server 702 to push an application command message ***to the push client 704 and thereby initiate a mobile terminal client trigger event in a mobile application 705*** from a plurality of such applications, program modules and user interface features on the terminal 705." Ex-1007, 21-22. According to Houghton, a "trigger event" message is a message that includes data for "launch[ing] or command[ing] a mobile application." *Id.*, 9.



Id., Fig. 7.

404. "The trigger mechanism may be between separate parts of a larger application or application suite, or may be an IP-triggered application launch and data transfer as in the case of mobile application launched by contact to a specified IP port." *Id.*, 22. The "push 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," which causes "pushing application launch commands, lazy application updates, and lazy application data updates through push clients 904 and 905 to mobile application 907 through local command and data path D (also Figure 5, arrow A)." *Id.*, 24.

405. In view of the above disclosures, in my opinion, a POSITA would have understood that Houghton discloses or suggests that the received network element messages (push messages from application server(s) 702/802) include message

content with data for the destination mobile application, as well as an identification of this application (based on the information "specifying which mobile application 407 from a plurality of such applications"). Moreover, a POSITA would have understood or found it obvious that the received push messages from the application server request delivery of message content to particular mobile terminals, and in particular, to the applications resident on those mobile terminals.

406. For the above reasons, Houghton-Munson renders obvious 1[c2]/15[c2].

vi. **1[d1]-1[d2]:** *a message buffer system including a memory and logic, 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, /* **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;*

407. Houghton-Munson renders obvious a message buffer system including a memory that buffers content from the received network element messages for which delivery is requested to a given one of the wireless end-user devices.

408. As I explain in the combination of Houghton and Munson (and incorporate here), in my opinion, a POSITA would have found it obvious in view of Munson to implement a message buffer in Houghton's push system to store received push messages on Houghton's push server. In more detail, and as explained with reference to 1[c1]/15[c1]-1[c2]/15[c2] (which I incorporate here), Houghton-

Munson render obvious that the push server receives network element messages in the form of push messages from application servers, where delivery of such messages is requested to one or more mobile terminals/wireless end-user devices.

409. The push server further includes a message buffer system that includes logic (as described with reference to 1[d3]/15[d2] and 1[d4]/15[d3] below and which I incorporate here) and, in view of Munson, would have also included memory that buffers content from the received push messages (for the reasons stated in the Combination of Houghton and Munson, above, which I incorporate here).

410. Indeed, Houghton discloses that push messaging services, like "SMS (Short Message Service)" and "MMS (Multi-media Messaging Service)" "store and forward" push messages. Ex-1007, 3, Figs. 3-4. Houghton further discloses that servers (such as push servers) "control and deliver messages and store messages when the mobile terminal is unavailable" and then later deliver those stored messages when "the terminal becomes available." *Id.*, 7.

411. Given (1) messages may be received by push server when the client device is unavailable and (2) Houghton's express disclosure of storing such messages until the user becomes available, a POSITA would have understood or found it obvious that the push server would include a buffer/memory for storing received messages.

412. Indeed, in the similar context of push communications, it was well-known, as taught in Lee, to store push messages "when the mobile terminal is not connected to the mobile internet service network" and "re-transmitting the stored push information when the mobile terminal is connected to the mobile internet service network" *See* Ex-1006, claim 14, Fig. 1 (showing a push information storage device 112 for storing push messages). Lee also contemplates that a scheduler 114 can "schedule and transmit the push information push_info or retransmit the push information push_info." *See id.*, [0034], Fig. 34.

413. Consistent with these well-known teachings of storing push messages in push server's memory/storage, Munson, which also enables push messages, discloses a "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 until all contents in a series arrive." Ex-1017, [0036]-[0037], Fig. 4. Munson's Figure 3 illustrates a structure for queuing messages in a Content Push Service 220. *Id.*, [0035]-[0036], Fig. 3.

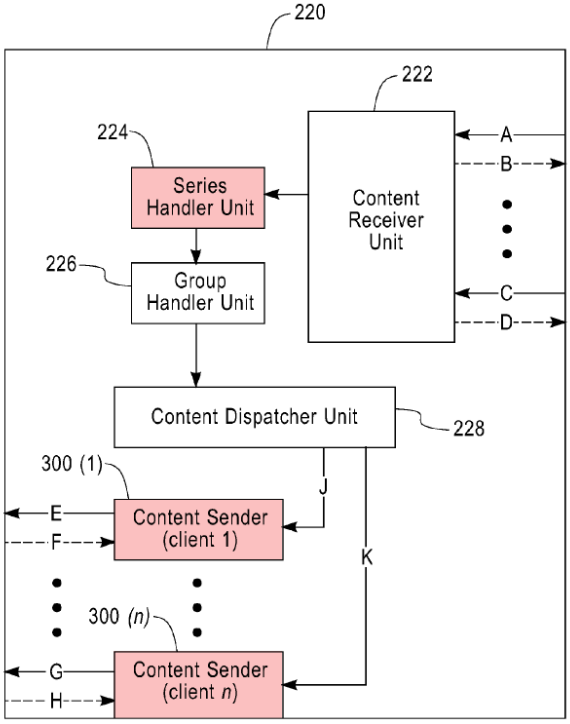


FIG. 3

Id., Fig. 3.

414. Munson additionally discloses a plurality of "Content Sender[s] 300" that each contain a "Queuer 320" and "Push Queues 330." Ex-1017, [0036], [0038], Figs. 3, 5. Munson discloses that when pushing messages, Content Senders 300 check "urgency level of contents … customer level … [and] dequeue policy." *Id.* Moreover, each "client device" (e.g., the Houghton-Munson mobile terminal) is assigned a Content Sender 300. *Id.* Munson's Figure 5 illustrates a structure for queuing messages in a Content Sender 300. Ex-1017, Fig. 5.

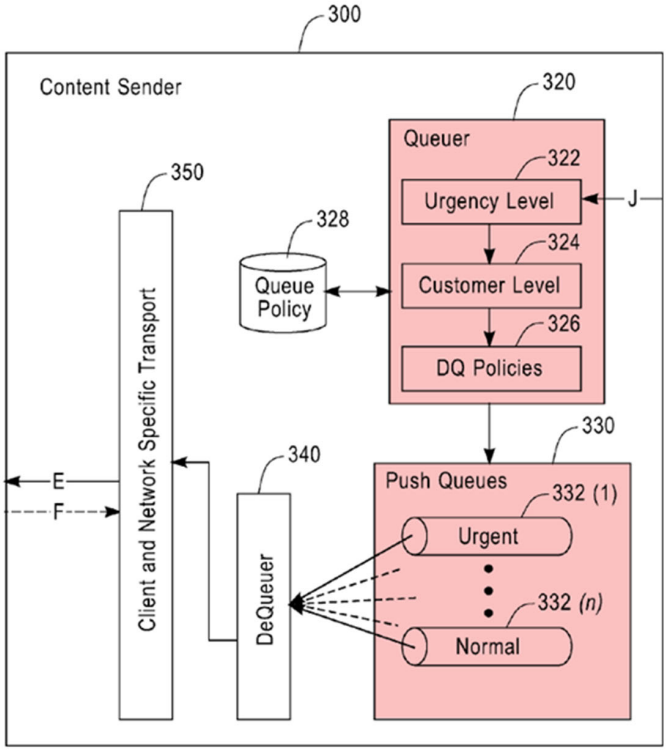


FIG. 5

Id., Fig. 5.

415. In view of the above-described teachings in Munson and Houghton, and for the reasons stated in the Combination of Houghton and Munson, above, which I incorporate here, a POSITA would have found it obvious to implement computer memory and programming to deliver messages from memory, consistent with Munson, as part of Houghton's push server to enable storage and later forwarding of messages to one or more mobile terminals/push client, e.g., upon occurrence of certain trigger events. Thus, in the Houghton-Munson push server, the received network elements messages would be stored in the buffer and associated

components, as described in Munson and as implemented in Houghton's push server.

416. Additionally, as described below (1[d3]/15[d2]-1[d4]/15[d3], which I incorporate here), Houghton's push server includes a message buffer system with logic that facilitates delivery of messages upon occurrence of one or more message delivery triggers.

417. For the above reasons, Houghton-Munson renders obvious 1[d1]-1[d2]/15[d1].

> ***vii.*** **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 /* **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*

418. As explained below, Houghton-Munson renders obvious the message buffer system including logic that determines when one of a plurality of message

delivery triggers for the given one of the wireless end-user devices has occurred, where the receipt of a message is not a message delivery trigger and one of the message delivery triggers is an occurrence of an asynchronous event with time-critical messaging needs (e.g., a user request for a received message or the occurrence of a transaction).

419. In more detail, Houghton contemplates the occurrence of trigger events to trigger sending push messages to the client devices. For example, Houghton discloses that the "***push message from the push server 401 may be triggered by any trigger event, local or remote, defined at the server. The trigger may, among others, include an alarm, notification, or measurement result received to the push server 401 from another device or system***." Ex-1007, 21; *see id.*, 14 (describing various trigger events), 21-22, 25-28 (describing various trigger events).

420. Houghton therefore discloses that the delivery of push messages can be triggered upon the occurrence of certain trigger events and that the trigger event can be "local or remote" and may include "an alarm, notification, or measurement result received" by the push server 401 from another device or system. *See id.*, 21. Thus, a POSITA would have understood or found it obvious that Houghton's system would implement message delivery triggers for triggering the delivery of messages to the wireless end-user devices.

421. Moreover, the triggers noted above—alarm, notification, or measurement result—are message delivery triggers and do not constitute a trigger that is based on receipt of the message by the push server.

422. Additionally, Houghton discloses or renders obvious that the message delivery triggers include an occurrence of "an asynchronous event with time-critical messaging needs," consistent with how this claim phrase is described in the '192 patent's specification. Specifically, and as explained with reference to this limitation in Ground 1A above (which I incorporate here), the '192 specification explains that an asynchronous event with time-critical messaging needs encompasses a "user request" for the delivery of the message or the occurrence of a transaction. Ex-1001 ('192 Patent), 38:50-64.

423. Consistent with this description, Houghton renders obvious a message delivery trigger that is an occurrence of an asynchronous event with time-critical messaging needs. Specifically, and as noted in the above-cited disclosures, the trigger event can be an alarm or notification received from another device.

424. Houghton further contemplates that message delivery can be adjusted based on whether the message to be delivered is time-critical or "non-time-critical." *See, e.g.*, Ex-1007, 20-21 (contemplating "time critical and the fastest available"), 23-24 ("[t]he push server 501 on becoming aware of such a situation may inform (arrow D) the application server 502 of the availability of a more desirable network

connection (C plus A) to the client and on behalf of the application server 502 or independently decide to push additional commands and data to the wireless terminal using this connection. Further, the push client in the broadband access point has the additional capability to route such information to the wireless terminal. ***The result benefits customers using the wireless terminal in that data is managed and updated automatically for their mobile applications in the most cost efficient manner for non-time-critical capabilities such as lazy application code update and lazy application data update***. While such alternative routing may also be achieved through normal IP routing and path cost without the presence in the broadband access point 509 of push-aware client software, ***the push server 501 and application server 502 would in this known-in-the-art implementation not be aware of when the low cost alternative route was available and thus able to at such time trigger the push of lazy application code and data installation and updates***."). A POSITA would have therefore found it obvious to modify message delivery based on the time criticality of the message and therefore, adjust message delivery and the associated trigger in this context.

425. Similar to Houghton, Munson discloses that its push system provides "***asynchronous content push[es]*** (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" (Ex-1017, [0040],

[0044]), thus rendering it obvious that a message delivery trigger could be "an asynchronous event with time-critical needs" as claimed. In my opinion, therefore, even if Houghton's triggers do not disclose this limitation, the combination of Houghton and Munson does because Houghton contemplates configuring the system for "any trigger event," which would have included those described in Munson as additional example triggers in a similar networking system. Houghton, 21.

426. Houghton's push client can also send a message based on a user request to the push server, requesting delivery of a push message. Specifically, "IP push command messages may be triggered" when the "push client makes" push server aware of certain "client-side events," which in turn trigger the push server to deliver push messages intended for particular mobile application(s) 406 residing on the mobile terminal. Ex-1007, 14, 21. For example, the "client-side events" including, among others, "***creation of photographs, video or other media***," "***video game actions or events***," and ***"messaging actions***" are triggers that "***push client makes the server aware of***" so that the server pushes a message to the mobile terminal and a particular application residing therein. *Id.*, 14. Houghton further discloses that the trigger events include "***remote application user*** or application server ***event***, ***remote user action***." *Id.*, claim 22.

427. Given the above disclosures, in my opinion, a POSITA would have understood or found it obvious that these client-side events are user requested events

or transactions that occur, e.g., as a result of transactions and events occurring on a mobile terminal, including user actions taken with respect to creating media, gaming events, or messaging.

428. A POSITA would have thus understood or found it obvious that such user actions or transactions trigger delivery of messages and therefore, constitute message delivery triggers that constitute an asynchronous event with time-critical messaging needs, as that phrase would be understood in the context of the '192 Patent.

429. Because the push server waits for these triggers to occur before it deliver the message to push clients, a POSITA would have understood or found it obvious that push server includes logic that determines when one (or more) of these message delivery triggers for the particular terminal/end-user device has occurred and if so, delivering the messages. In my opinion, POSITAs would have therefore found obvious, in view of Houghton or Houghton-Munson, to adjust message delivery based on the message's time criticality and therefore, adjust message delivery and the associated trigger on an asynchronous basis.

430. For the above reasons, Houghton-Munson renders obvious 1[d3]/15[d2].

viii. **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. /* **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.*

431. Houghton-Munson renders obvious that, upon determining that one of the message delivery triggers has occurred (see above analysis for 1[d3]/15[d2]), the logic further to supply one or more messages comprising the buffered content (see above analysis for 1[d1]-1[d2]/15[d1] relating to storage of received push messages) 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.

432. As I described in 1[a]/15[a]-1[b]/15[b] (which I incorporate here), Houghton's push server would have included a transport services stack that facilitates a secure message link between the push server and the push client (on the mobile terminals). Houghton further discloses, in view of Munson and as described with reference to 1[d1]-1[d2]/15[d1] (which I incorporate here), to store the received push messages in push server's memory. In my opinion, as I explained in the

Combination of Houghton and Munson, a POSITA would have appreciated Munson's content senders operate similarly to Houghton's routing to transmit messages from the buffer to the client device. Ex-1017, [0036]-[0037].

433. In the combination, the push server would include logic, and the stored push messages (and the associated data/content) would be delivered to the push client upon the occurrence of one or more message delivery triggers, as described with reference to 1[d3]/15[d2] (which I incorporate here).

434. Specifically, Houghton discloses "intelligence" that "route[s] messages received on behalf of the wireless terminal to and from the server." Ex-1007, 14. Such message delivery would happen using the secure message link already established/maintained between the push server and the push client of the mobile terminal. *Id.*, claim 1 ("the push server sends, in response to predetermined trigger events, at any time push application installation, updates, commands or data messages to be directed to a push application or dynamically loaded module or user interface feature from among a plurality of such applications, modules and features by means of said client-initiated or client-opened and client-maintained permanent connection").

435. For the above reasons, Houghton-Munson renders obvious 1[d4]/15[d3].

b. Claim 5

436. Houghton-Munson renders obvious the message link server (as described with reference to claims 1 and 15 above) and further renders obvious 5[a]-5[c], as described below.

> ***i.*** **5[a]:** *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,* **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,*

437. Houghton-Munson renders obvious that push server's transport services stack (as described in 1[a]/15[a] above) receives, over each of the respective secure message links, upload messages forwarded by the respective device link agents (e.g., one or more push clients) from at least a subset of the device software components (e.g., one or more applications). Moreover, as I describe in 1[c2]/15[c2] (which I incorporate here), Houghton discloses providing an application identifier in (upload) messages.

438. In more detail, Houghton discloses that push client 405 "at any time accept[s] ***data D from each of a plurality of mobile applications 406 and direct[s] such to the push server 401***, possibly for redirection through push server API to other servers, mobile terminals, push clients and computing devices." Ex-1007, 21;

see id., 26 ("***the client 405 may also send a message to the server 401 (step 633),*** such as a command execution acknowledgement or result"). The "messages received on behalf of the wireless terminal" may be routed "to and from the server." *Id.*, 14. The message's "data packet" could include "information specifying which mobile application 406 from a plurality of such applications" it will be directed to. *Id.*, 21.

439. Moreover, given that Houghton's push server includes a transport services stack (as described with reference to 1[a]15[a] above and which I incorporate here), including API or IP-based connections between the push server and the application server—and may encrypt those connections—and includes similar connections between mobile terminals and the push server, a POSITA would have understood or found it obvious that the push server's transport services stack would receive, over the secure message link (as described with reference to claim 1), messages including data (i.e., upload messages) forwarded by the push client(s) (i.e., respective device link agents) from at least a subset of the applications 805 (i.e., device software components).

440. Additionally, as described above, each such message would be directed to an application server, and thus, a POSITA would have understood or found it obvious that each such message would identify the particular application server (i.e., network element) to which the initiating software application (i.e., device respective software component) requested delivery.

ii. **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.*

441. As described with reference to 5[a]-5[b] and claims 1 and 15 (which I incorporate here), Houghton's push server (i.e., the network server system) includes interfaces that facilitate communication with network elements (i.e., interface to a network), including mobile terminals and application servers, to deliver messages and associated data (i.e., content from the upload messages) to one or more application servers (i.e., the respective identified network element, of which there can be more than one).

442. Moreover, as described with reference to 5[a]-5[b] (which I incorporate here), the push server uses an API (application programming interface) or IP connection to facilitate communication between the push server and the application server, and a POSITA would have understood that this interface/network connection would have been used for communicating the upload messages to the application server(s) from the push server.[7]

443. For the above reasons, Houghton-Munson renders obvious 5[a]-5[c].

[7] I note that claim 5 uses the term "network server system" that does not appear elsewhere in the claims. However, I have assumed that the "network server system" refers to the "message link server" that appears in claim 1.

c. ***Claim 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.*

444. Houghton-Munson renders obvious that at least one of the one or more messages for delivery by the transport services stack comprises multiple identifier/data pairs.

445. Specifically, as I described in 1[a]/15[a] (which I incorporate here), Houghton's push server includes a stack of secure protocols for API or IP-based connections between the push server and the application server to establish a connection between mobile terminals and the push server.

446. As I also described in 1[b]/15[b] and 1[c2]/15[c2] (which I incorporate here), push messages would be received by the transport services stack implemented as part of Houghton's push server and from different network elements (e.g., application server or other push client resident on a respective mobile terminal), and each such message intended for an application executing on one of the mobile terminals would include data and corresponding identification of the intended software application.

447. Based on these teachings, a POSITA would have recognized or found it obvious that messages received from the push server would have been directed to one or more applications from among the "plurality" of applications ("***multiple identifier/data pairs***") on a user's mobile terminal because Houghton discloses

various "actions" that occur across applications when receiving a push message. Ex-1007, 21. Specifically, the message's "data packet" may include "information specifying which mobile application 406 from a plurality of such applications" it will be directed to. *Id.*, 21. This is corroborated by Houghton, which discloses that, in some cases, the "packaging of mobile applications involves combining multiple applications delivered in a single bundle" (e.g., a "bundle" or "suite" of applications would receive messages packaged together (*id.*, 12) – a message containing "***multiple identifier/data pairs***," as claimed.

448. This therefore demonstrates that the receipt of multiple identifier/data pairs in the one or more messages for delivery by the transport services stack of the push server.

449. For the above reasons, Houghton-Munson renders obvious claim 6.

d. ***Claim 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.*

450. In Houghton, "the mobile terminal is provided with means, such as software, which ***initiates and maintains contact with the server using Internet technologies***" and this "***mobile-initiated permanent IP connection*** allows the server to pass or push messages from outside the operator's firewall to the mobile terminal at any time." Ex-1007, Abstract, 11 (disclosing a "client-initiated" connection); *see*

id., claim 1, 25 (describing push client initiating and maintaining an IP connection link with push server).

451. Additionally, as described in 1[a]/15[a] (which I incorporate here), SSL/TSL is used for secure messaging between push client and push server. Thus, a POSITA would have found it obvious for the SSL/TSL-based communication link between push client and push server to be initiated by the client device. Indeed, before the Critical Date, it was well-known for SSL-based communication between a mobile terminal and a server to be initiated by the client device. *See, e.g.*, Ex-1012, 29:31-30:24 (describing that SSL/TLS requires a handshake phase and a data transfer phase, and in the handshake phase, the ***client side initiates a handshake*** by sending the server a ClientHello message).

452. Given that the push client logs into the push server, facilitates communication between these entities using the established IP connection, and uses SSL-protocols that are initiated at the client side/mobile terminal, a POSITA would have found it obvious that the push client (i.e., the device messaging agent on at least one of the wireless end-user devices) initiates the respective secure Internet Data message link (message link on which data is sent over the Internet) to push server's transport services stack (as described with reference to 1[a] above).

453. For the above reasons, Houghton-Munson renders obvious claim 7.

e. ***Claim 9****: The message link server of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

454. Houghton-Munson renders obvious that one of the message delivery triggers is the expiration of a periodic timer.

455. Houghton contemplates multiple message delivery triggers (Ex-1007, 14, 21) and additionally discloses a periodic message (which would obviously be triggered at "the expiration of a periodic timer" as claimed) that ensures that devices in the communication path (i.e., push client at mobile terminal and push server) "do not time expire the connection," suggesting that the periodic messages in Houghton could recur regularly. *Id.*, 26, 19 ("Use of a periodic message from the client to the server or server to the client to the effect that client socket, server socket, intermediate Network Address Translation (NAT) devices, intermediate firewalls, and intermediate gateway devices do not time expire the connection."), 11 ("In an embodiment of the invention, the mobile terminal maintains contact with the server by periodically testing the connection with a suitable keepalive message, a 'ping', which resets timers in intervening operator stateful firewalls and/or NAT (network address translation) firewalls. The mobile terminal may also automatically re-establish the connection when the connection gets broken and adjust its connection settings based on past experience with failure to connect and such broken connections in a given mobile network.").

456. Houghton explains that the "periodic message timings" can be "measured in seconds, minutes or hours" and can be "longer or shorter…to match the combined needs of all such network devices and protocols." *Id.*, 19-20, 26. Additionally, the timing between periodic messages can be "adjusted based on the success or failure of earlier messages," also suggesting that these messages could recur at regular intervals. *Id.*, 19. Munson also discloses that "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 to take advantage of lower rates." Ex-1017, [0044]. As I explained previously in 1[d3]/15[d2] (and incorporate here), a POSITA would have found it obvious to configure Houghton to implement Munson's triggers. A POSITA would have understood or found obvious that "schedule[d]" pushes during off-peak hours for maintenance would be triggered by the expiration of a regularly-repeating timer.

457. Given Houghton's disclosures of delivery triggers and Munson's use of periodic timers for message delivery, in my opinion, a POSITA would have been motivated and would have had a reasonable expectation of success in implementing Munson's teaching of using periodic timers (and expiration of the same) as a message delivery trigger, within Houghton's system that already contemplates using periodic timers. This combination would have been sufficient to cause message delivery. A POSITA would have been motivated to use periodic timers as a message

delivery trigger, as contemplated by Munson, to enable more efficient use of network resources and avoid repeated message requests that can impact performance of network communications.

458. For the above reasons, Houghton-Munson renders obvious claim 9.

f. ***Claim 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.*

459. Houghton-Munson renders obvious that the period of the timer is fractionally shorter than a maximum data message interval after which the secure message link is taken down.

460. Specifically, Houghton discloses "the push client 405 may send a periodic message…from the client 405 to the server 401 to effect that…[the network devices] do not time expire the connection. The period of time…between such periodic messages is set to match the combined needs of all such network devices and network protocols". Ex-1007, 26. Houghton discloses that the periodic messages are sent to ensure that devices in the communication path (i.e., push client at mobile terminal and push server) "do not time expire the connection," suggesting that the periodic messages in Houghton could recur regularly. *Id.*, 26, 19-20. A POSITA would have understood that this period sets a "maximum data message interval beyond which the secure message link is taken down" as claimed because the

purpose of the message is to *prevent* expiration of the connection; failing to receive a period message will therefore terminate the connection.

461. In my opinion, a POSITA would have understood that the period to send a regularly recurring timer for triggering message delivery (which I again describe in claim 9 and incorporate here) would be some degree shorter than the recurring time for Houghton-Munson's periodic messages (e.g., the claimed "maximum data message interval beyond which the secure message link is taken down"). If the periodic polling/message delivery was set to a longer period than the keep-alive timer period, the connection may be terminated before the periodic polling/delivery message could be sent, thus requiring a new connection to be established. And because the '192 Patent does not specify what constitutes "fractionally shorter," in my opinion, any degree shorter can be considered "fractionally shorter," as claimed, since it would be some "fraction" of the duration of the keep-alive timers.

462. In my opinion, because Houghton discloses both a periodic timer *and* periodic (connection) messages, a POSITA would have understood that Houghton discloses a periodic message set for a period of time that is shorter than a connection timeout after which the client and server are disconnected.

g. ***Claims 11-12****: **[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. **[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.*

463. As described below, Houghton provides multiple examples of message delivery triggers that include "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," where the transmission is a "request received from the given device link agent."

464. For example, Houghton describes that "client-side events" can result in "pushing application commands." Ex-1007, 14, 21, 16 ("Figure 4 illustrates an exemplary implementation of the system ***for pushing application commands to a mobile terminal based on*** server side and ***client side events*** …."), Fig. 4. Trigger events include "messaging actions" or a "notification." Ex-1007, 14, 21. Houghton's "client" notifies the "server" of a "terminal event" that triggers a "return message." *Id.*, 20-21. Houghton further discloses that "[i]f 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. Specifically, Houghton also describes that the

"client" can notify the "server" of a "terminal event" that triggers a "return message." Ex-1007, 20-21.

465. Additionally, Houghton contemplates that a "connectionless protocol, such as UDP/IP," may be used "when the ***client 405 notifies the server 401*** of a[] terminal event, user interface event or application event" and the ***push server "in return 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." Ex-1007, 20-21.

466. In the context of established IP connections, the "push client 405 may monitor the local resources for a state change (a trigger event) in step 641. If a state change of other event is detected in step 642, the push client 405 may send a message to the server 401 (step 643)." *Id.*, 26.

467. Moreover, as described above with reference to claim 9 (and which I incorporate here), an example message trigger includes periodic polling to ensure that devices in the communication path (push client at mobile terminal and push server) "do not time expire the connection," which a POSITA would have recognized or found it obvious as being akin to a heartbeat message generated by the given device link agent. *Id.*, 19, 21, 26; Ex-1001 ('192 Patent), 38:50-63, 70:6-71:16.

468. For the above reasons, Houghton-Munson renders obvious claims 11-12.

h. ***Claim 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.*

469. Houghton-Munson renders obvious that one of the message delivery triggers is the receipt of a particular network element message from one of the network elements (e.g., a message from a particular application server). *See, e.g.*, Ex-1007, 21-22 ("Figure 7 illustrates an example of [a] 'COMMAND PUSH' procedure ***wherein the push server 701 is triggered by a trigger event as for example from an application server 702 to push an application command message to the push client 704*** and thereby initiate a mobile terminal client trigger event in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705."), Fig. 7.

470. For the above reasons, Houghton-Munson renders obvious claim 13.

I. Ground 2B: Houghton, Munson, and TS-23.140 (Claims 1, 15)

471. If Houghton-Munson does not render obvious claims 1 and 15—and specifically that applications must register/be authorized or that the received message includes an application identifier—in my opinion, it would have further been obvious to implement these features as taught by TS-23.140.

472. TS-23.140 teaches that applications must register before being granted access to the MMS network, during which "the application provisions an MMS User Agent or an MMS VAS Application with its application identification value and may

negotiate with the MMS User Agent or MMS VAS Application the details (amount and format) of information to be exchanged between the two entities." Ex-1004, 54-55. In other words, the registration process is intended to ensure the messages the application intends to send using the MMS network are compatible with the network, and that the network will be able to transmit/receive application-specific messages as intended. TS-23.140 also discloses that application-specific messages sent by an MMS User Agent or an MMS VAS Application includes an "application identifier of the destination application." Ex-1004, 55. The application identifier is used to direct application-specific messages to the intended target. *Id.*

473. In my opinion, a POSITA would have first been motivated to incorporate an application-registration process like TS-23.140's into Houghton-Munson's system to ensure each application is configured to transmit messages compatible with the network and ensure successful message deliveries. Ex-1004, 54. To implement this feature, for instance, an application would negotiate with a push client to establish: (1) the application's identity to the network; (2) the formats, procedures, and protocols to be used when transmitting incoming/outgoing messages to/from the application through the push client. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification because it merely involves well-known, negotiated signaling between the registering application and existing messaging-network element (such as Houghton-Munson's

push client). In fact, TS-23.140 leaves the specific implementation of the registration process to those in the art to customize, which is an acknowledgement that those in the art knew how to implement specific registration processes.

474. After an application has registered and become authorized to access the network, in my opinion, a POSITA would have been further motivated to include application identifiers in application messages sent through Houghton-Munson's system, like TS-23.140 does, to ensure those messages are directed to the intended application, or to verify a message received by, e.g., an application server was issued by a compatible application. To analogize, if a user device is thought of as an apartment building, and each application on the device is a different apartment, it is like putting the apartment number on a mailed letter; you would not simply send letters with the building number hoping it reaches the right apartment in the building. Nor would some company of authority (akin to an application server) accept and execute based on the contents of mail (like application-message contents) from senders who fail to identify themselves. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification as well because the combination merely involves incorporating an express "application identifier"—typically a small text string—into the message. *See* Ex-1004, 55 ("The format of the application identifiers' value shall be text string.").

475. Moreover, if a combination with TS-23.140 is necessary, in my opinion, the combination would also teach all other Challenged Claims set forth in Grounds 2A, and 2C-G for the reasons states therein, in combination with TS-23.140. The remaining combinations involve discrete teachings separate from an application identifier, so any further combinations would, in my opinion, not impact the inclusion of an application identifier like TS-23.140 in Houghton-Munson.

J. Ground 2C: Claims 2, 3, and 12 Are Rendered Obvious by Houghton, Munson, and Shen

1. ***Claim 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.*

476. Houghton-Munson-Shen renders obvious 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.

477. As described with reference to 1[a]/15[a] and 1[d4]/15[d3] (which I incorporate here), Houghton discloses that messages would be supplied to the push server's transport services stack (Houghton's push server protocols) for delivery on the secure message link (e.g., SSL and HTTPS) between the push server and the push client (resident on a particular user device/mobile terminal). Although

Houghton discloses encrypting the connection between the push server and push client, it does not appear to disclose encrypting the messages themselves.

478. Shen, however, explains that the store and forward functionality like Houghton's (Ex-1007, 3, 7)—without encrypting the messages—exposes a "potential security problem" given that such systems are "not an end-to-end solution …." Ex-1014, [0004].

479. To overcome this "security problem," Shen's invention in one embodiment discloses that "an authentication and key management module 502 is added in the MMS gateway to distribute symmetric keys to users and a certificate management module 504 is added in the user device" for "encrypting MMS messages" as shown below (*id.*, [0054], [0059]):

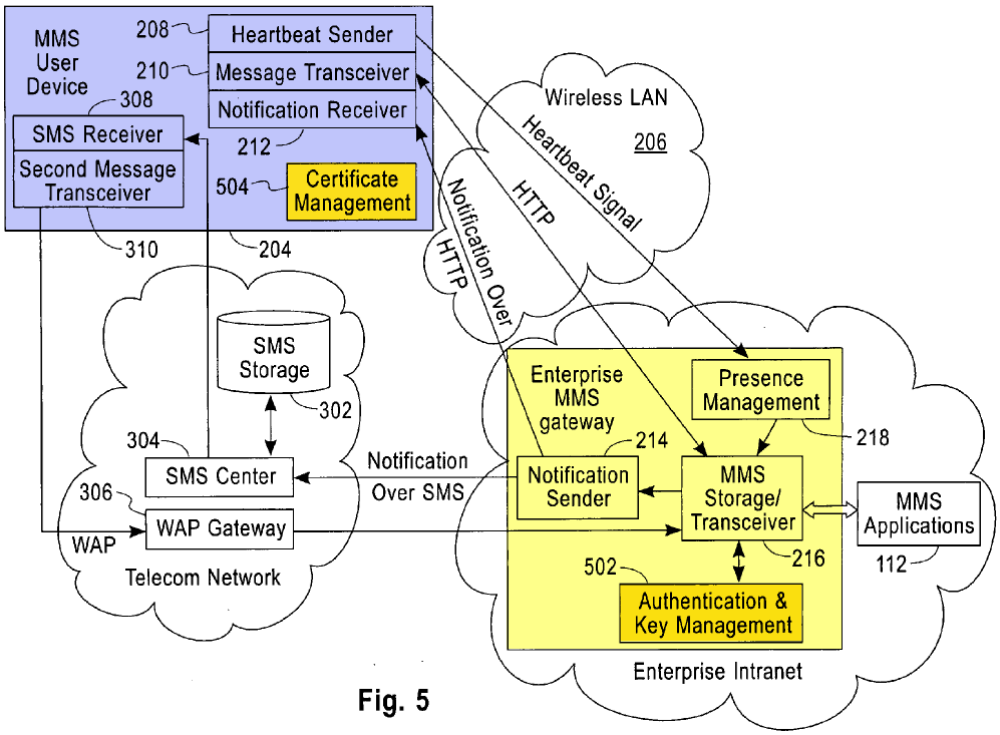


Fig. 5

Id., Fig. 5.

480. Using these modules, Shen discloses generation of a symmetric key that is then used by the MMS Relay/Server to encrypt data communications transmitted to the MMS User Device/Agent of user devices. *Id.*, [0055]-[0060]; *see also id.*, [0029]-[0034].

481. In my opinion, a POSITA would have been motivated to combine the teachings of Houghton-Munson and Shen to achieve a messaging system "having improved security," resulting from the "provi[sion] [of] an end-to-end security solution for MMS applications," which is particularly beneficial in the context of "enterprise applications." Ex-1014, [0017], [0021] (describing additional

advantages stemming from Shen's invention); Ex-1007, 22 (describing applicability of push messaging in the context of enterprise applications). Specifically, a POSITA would have been motivated to implement an "authentication and key management module 502" at the push server that generates and "distribute[s] symmetric keys to users" and using this symmetric key to encrypt data transmitted to "the user device," as contemplated by Shen. Ex-1014, [0054]-[0060].

482. Moreover, it was well known before the Critical Date for push servers to encrypt content of messages sent to push clients/client devices. Chou is one example of a push message system that uses a "WAP gateway" (e.g., a network message server) that encrypts content received from a "content server" for transmission to a "mobile client" that then "decodes the response." Ex-1009, 6:15-46. Rakic and Shen provide additional examples of a push server encrypting message traffic. Ex-1015, [0065]-[0068]; Ex-1014, [0054]-[0060]

483. Combining the above-described teaching of Shen with those of Houghton-Munson would have been obvious to a POSITA at least because such a combination would have merely involved combining prior art elements (a messaging network and message encryption) according to known methods (using encryption schemes and key generators) to yield predictable results (encrypted messages). Similarly, the combination would have amounted to using a known technique

(message encryption) to improve similar devices (messaging networks) in the same way (securing content).

484. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification because Shen describes applying encryption to a similar messaging environment as Houghton and communication between push servers and push clients, (Ex-1014, [0017]), such that a POSITA would have found it straightforward to modify Houghton-Munson's push server to implement Shen's security modules.

485. Additionally, the resulting system's elements would each perform functions they were known to perform prior to the combination—i.e., Houghton-Munson's system would continue to facilitate the communication of push messages between push servers and push clients, and Shen's teachings (as implemented in combination) would facilitate encryption by the push server for messages that are then sent to a push client.

486. Upon combining Houghton-Munson and Shen, a POSITA would have appreciated that Houghton-Munson-Shen provides an encrypt function to encrypt messages supplied to the components implementing Houghton's push-server protocols for delivery on the encrypted link maintained between the push server and push client to "encrypting MMS messages with the shared secret key." Ex-1014, [0059].

487. For the above reasons, Houghton-Munson-Shen renders obvious claim 2.

> ***2.*** ***Claim 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.*

488. Houghton-Munson-Shen renders obvious 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.

489. As described with respect to claim 2 (which I incorporate here), Houghton-Munson-Shen renders obvious encrypting messages and transporting the encrypted messages to a particular push client on a mobile terminal. Additionally, as explained with reference to 1[a]/15[a] (which I incorporate here), the combination renders obvious encryption on the transport services stack in view of the above-described teachings of SSL, HTTPS, and IPSec on the transport/communication link between the push server and the push client (i.e., encryption on the transport services stack and IP layer encryption). As Ozaki confirms, IPSec is encryption at the IP layer. Ex-1005, [0019]-[0021], [0008]-[0010]; *see also* Ex-1007, cl. 10 (also discussing tunneling, which is a technique to secure network packets).

490. For the above reasons, Houghton-Munson-Shen renders obvious claim 3.

> ***3.*** ***Claim 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.*

491. In my opinion, to the extent Houghton does not disclose a heartbeat message, Houghton-Munson combined with Shen also renders obvious the use of a heartbeat message as a trigger.

492. As I explained in claim 9 (and incorporate here), Houghton discloses using a "periodic message" as a message-delivery trigger to keep a connection alive, and monitors for when the client (405) has become disconnected. Ex-1007, 19-20, 26-27. Although Houghton does not expressly describe this feature as a "heartbeat message," in my opinion, a POSITA would have understood that it is, as it performs identical functions as what those in the art called a "heartbeat message."

493. If, however, Houghton's "periodic message" cannot be considered a heartbeat, in my opinion, a POSITA would have been motivated to implement Houghton's "periodic message" as a heartbeat in view of Shen. For instance, Shen's *heartbeat message* was one well-known and widely implemented way to indicate the reachability of the device (in this case, to receive stored messages). Ex-1014, [0033]-[0034]. They were utilized so widely because they were known to be low-bandwidth (compared to transmitted messages themselves) and reliable mechanisms

to indicate a device is connected and available for network transmissions (and in this case to trigger message delivery). *See, e.g.*, Ex-1043, [0062] (providing an example "64-byte link heartbeat (HB) message").

494. For these reasons, even if Houghton's periodic message cannot be considered a heartbeat message, in my opinion, replacing one with the other 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). A POSITA would have had a reasonable expectation of success in making this modification because Houghton and Shen already provide for similar signaling. *Id.*, [0034].

495. Upon making the combination, a POSITA would have understood that Houghton-Munson-Shen teaches a message-delivery trigger comprising the transmission of a heartbeat message like Shen's, generated by the push client (405).

K. Ground 2D: Claim 4 Is Rendered Obvious by Houghton, Munson, and Ellison

1. ***Claim 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.*

496. Houghton-Munson-Ellison renders obvious executing the device link agent in a secure execution environment, with applications that interact with the device link agent operated outside of the secure execution environment.

497. As I explained previously in 1[a]/15[a] and 1[d4]/15[d3] (and incorporate here), Houghton discloses supplying messages through components implementing secure push-server protocols (e.g., SSL and HTTPS) for delivery over a connection between push server and a mobile terminal's push client. Ex-1007, 16, 21, Fig. 4. Houghton does not, however, appear to disclose implementing the push client in a secure environment.

498. Ellison is just one example of prior art that uses secure environments to inhibit attacks from malicious software. Ex-1019, Abstract, 1:16-51. To inhibit attacks, Ellison discloses both an "isolated execution mode" (depicted in red below) where "access … is restricted" and a "normal execution mode" that "operates in a non-secure environment, or a normal environment" (depicted in green below) "without the security features provided by the isolated execution mode." *Id.*, 4:65-5:1, 6:1-26, 8:25-32, Figs. 1A, 1C. Additionally, Ellison's teachings enable "normal" execution, for instance, for applications "***outside of the secure execution environment on that device***." *Id.*, 4:65-5:1, 6:1-26, Figs. 1A, 1C.

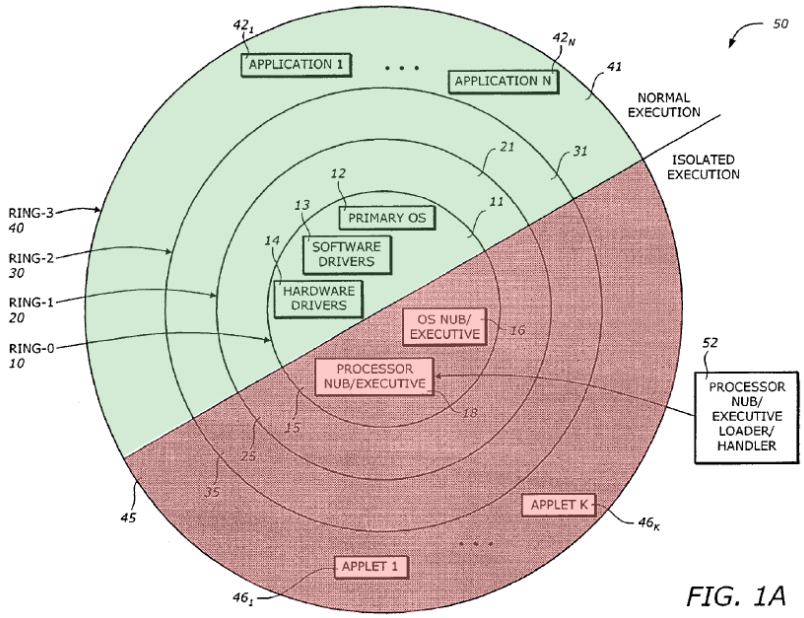


FIG. 1A

Ex-1019, Fig. 1A.

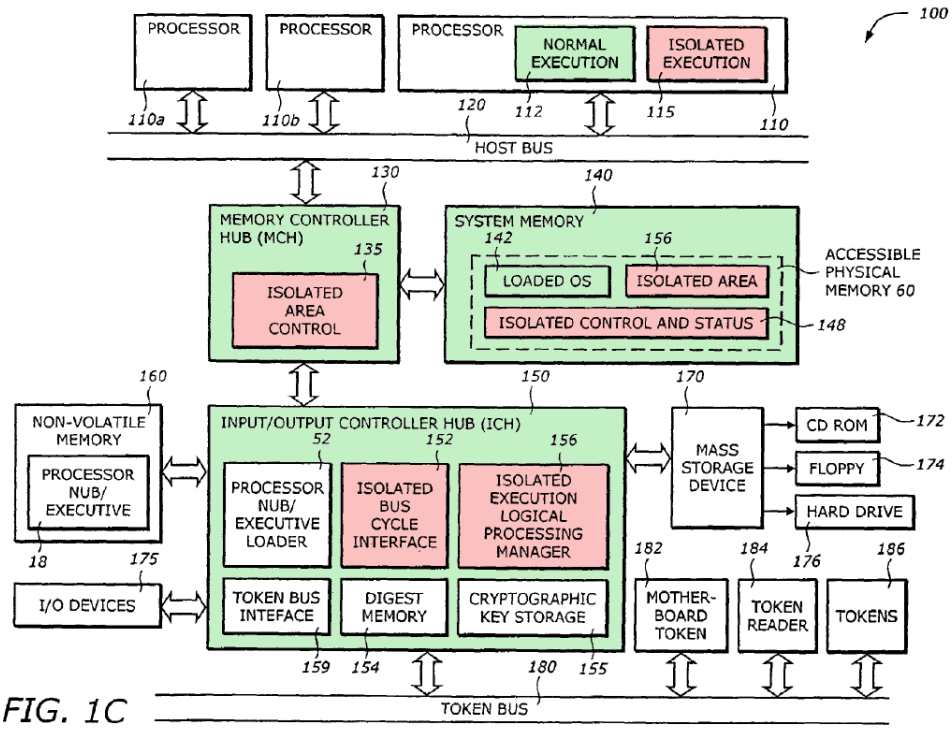


FIG. 1C

Ex-1019, Fig. 1C.

499. Ellison further discloses an "isolated area 70" which is "[a] memory area that is defined by the processor 110 when operating in [an] isolated execution mode." Ex-1019, 6:1-26, 8:25-32, Figs. 1A-1C, 2. Ellison refers to this technique, illustrated below in Figure 2, as a "secure platform." *Id.*, 6:1-26, 8:25-32, 8:66-9:62, Figs. 1A-1C, 2.

500. Ellison further teaches that the secure platform includes a "key generator 240" that "generates a key operating system nub key (OSNK) 203" which is then supplied to "trusted agents." *Id*., 8:66-9:6, Fig. 2. One example of a trusted agent is a "usage protector 250" that "uses the OSNK 203 to protect the usage of [a] subset 230." *Id*., 9:28-40, Fig. 2. Ellison's usage protector 250 uses a hashing function with the subset 203's ONSK 203 to determine if the subset 203 has been altered following changes (e.g., reads or writes to the subset 203), thereby providing "protection against unauthorized reads, and detection of intrusion, tampering or unauthorized modification." *Id*., 9:47-62.

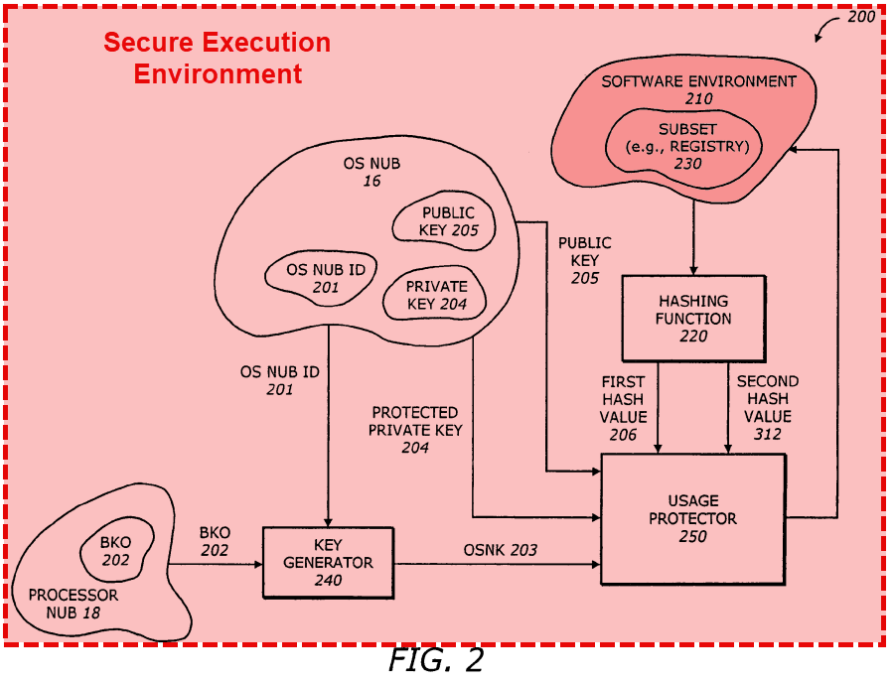


FIG. 2

Id., Fig. 2.

501. In my opinion, a POSITA would have been motivated to combine the teachings of Houghton-Munson and Ellison (and specifically to incorporate Ellison's hierarchical, nub-based normal and isolated execution environments into Houghton-Munson) to improve the security of Houghton's mobile terminal. *See, e.g.*, Ex-1007, Figs. 3-4, 8; Ex-1019, 4:63-5:10, 8:25-32, 8:66-9:6, Figs. 1A, 2. Such a combination would ensure that malicious applications could not easily gain access to stored message content after the encryption used during transmission had been removed by the intended user device.

502. A POSITA would have recognized that Ellison's secure platform would have improved the security of Houghton's push message system by improving

device security (i.e., entry points to the network). Ex-1019, 8:25-32, 8:66-9:6, 9:28-62, Fig. 2.

503. The increased confidence in device and/or network security provided by Ellison would have enabled a POSITA to Houghton's push service to additional application servers and clients. *Id.*, 8:25-32, 8:66-9:6, 9:28-62, Fig. 2.

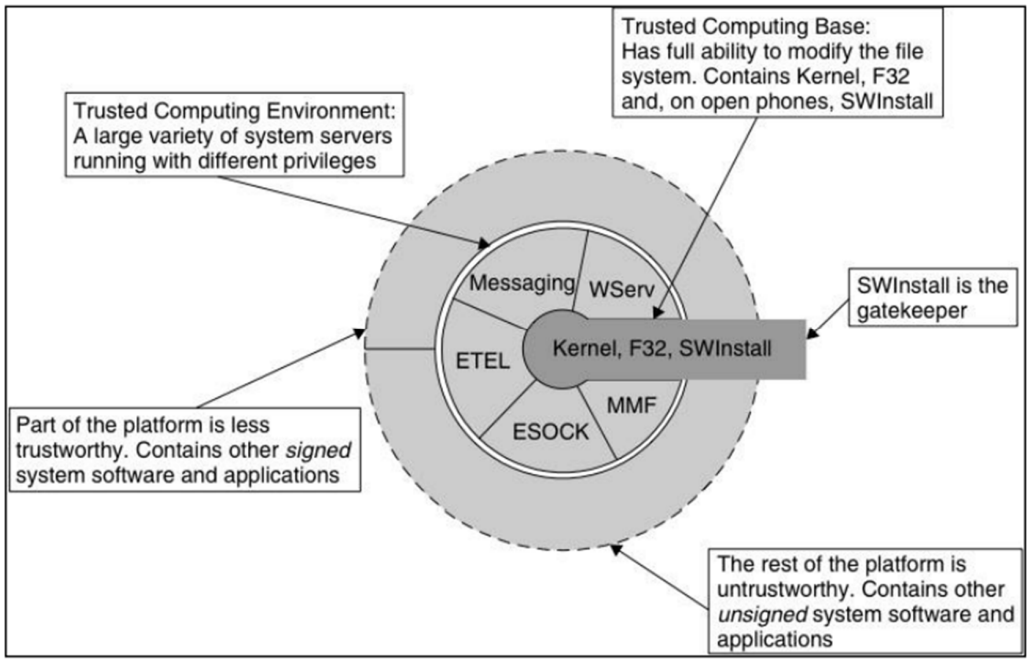
504. Moreover, device level security improvements are an efficient, beneficial option for improving network security, which would have been a natural solution for Houghton's system, which is concerned with secure messaging in the push messaging context. *Id.*, 8:25-32, 8:66-9:6, 9:28-62, Fig. 2.

505. In my opinion, combining the teachings of Houghton, Munson, and Ellison would have been obvious at least because such a combination would have merely involved combining prior art elements (computer messaging networks and secure computer executing environments) according to known methods (software/hardware implementations) to yield predictable results (select computer programs executing in secured environments).

506. In my opinion, a POSITA would have also had a reasonable expectation of success in incorporating Ellison's security techniques into Houghton's mobile terminals because Houghton contemplates security mechanism to secure its messaging systems, including messaging around application data, and Ellison describes that its techniques can be implemented in "computer system[s]" that

contain a "processor" similar to the network described in Houghton-Munson. Ex-1019, 5:11-16. A POSITA would have recognized or found it obvious that the examples of "mobile terminals" provided in Houghton include a "processor" capable of implementing Ellison's techniques as it was known in the art that mobile terminals contained processors. Ex-1007, 16; *see also* Ex-1019, 8:25-32, 8:66-9:6, 9:28-62, Fig. 2.

507. Moreover, in the context of mobile devices and their associated operating systems, it was well-known before the Critical Date to implement more secure systems, with rings or tiers of protection around different device components. *Id.*, 2:41-3:31, Fig. 1A. For example, in the early 2000s, the Symbian mobile operating system (B. Li, et al., *Symbian OS platform security model*, LOGIN: Magazine (Aug. 2010)—Ex-1033; Simon Higginson, Platform Security Concepts, in SYMBIAN OS PLATFORM SECURITY: SOFTWARE DEVELOPMENT USING THE SYMBIAN OS SECURITY ARCHITECTURE, 17-41 (Craig Heath ed., 2006)—Ex-1045) separated processes into three "Trust Tiers": The Trusted Computing Base (TCB), The Trusted Computing Environment (TCE), and Applications, as illustrated below:



Ex-1045, 24 (Fig. 1)

508. As shown above, the first two layers/tiers shown were primarily dedicated to components of the Operating System with relatively higher operating privileged, followed by the Applications tier with applications having relatively lower privileges. Ex-1033, 3-5, Fig. 1. The first two tiers included the Trusted Computing Base and the Trusted Computing Environment. *Id.*, Fig. 1. As shown, the Trusted Computing Environment includes, among other services, a Messaging service. *Id.* The Application Tier could be further subdivided into Signed and Unsigned applications. *Id.* Signed applications could interact with the TCE and have this layer execute higher privileged requests, whereas Unsigned applications could not access that same functionality. *Id.*, 4, Fig. 4.

509. Thus, it was well-known for operating systems to include the messaging services and clients/agents (e.g., push clients) responsible for provision of those services, within the trusted computing environment, and for applications (including those that would be intended recipients of those messages) to be tiered/organized outside this environment.

510. Given that Houghton contemplates mobile terminals with a push client that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging) from other applications/components of the environment, and Ellison's own disclosures confirming the same, a POSITA would have been motivated to similarly enable such improved security within the push messaging environment in Houghton by implementing Ellison's teachings of secure environments, such that the push client is part of the device's secure execution environment and enables communications with applications that reside outside this trusted/secure execution environment. Such a combination would have been straightforward to implement with a reasonable expectation of success given the well-known use of such techniques in the art and the recognition and use of security/encryption already contemplated in Houghton.

511. Specifically, a POSITA would have understood that Houghton's push client would operate in an isolated, secure environment that "is protected by both the

processor and chipset" (Ex-1019, 3:32-48), to reduce the push client/user's messages exposure to attack. Applications that interact with the push client would operate in normal or separate isolated environments to restrict the application(s) access to the push client. Ex-1019, 2:55-61 (noting applications operate in "Ring-3"—"the lowest level of the hierarchy" with the "least privilege").

512. In this manner, Houghton-Munson-Ellison renders obvious that the push client executes in a secure execution environment and one or more of the applications to which push messages are routed executes outside of the secure execution environment.

L. Ground 2E: Claim 8 Is Rendered Obvious by Houghton, Munson, and Rakic

1. ***Claim 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 se-cure message link messages via the message link server.*

513. For the reasons described below, Houghton-Munson in view of Rakic renders obvious that message link server (e.g., push server) comprises a secure server (e.g., a secure push message server) to provide secure authorization signatures to the given one of the wireless end-user devices (e.g., mobile terminals), the secure authorization signatures indicating the authorized software components (e.g., applications receiving application data that have previously undergone an

application registration process) that are to receive data from secure message link messages via the message link server.

514. As I previously explained in 1[b]/15[b] and 1[c2]/15[c2] (and incorporate here), Houghton discloses push clients operating on mobile terminals to receive messages over a secure connection via a push-server, including to/from various applications. *See, e.g.*, Ex-1007, 19-21. And as I further explained in 1[b]/15[b], a POSITA would have found it obvious that Houghton-Munson's mobile applications would register with the push client and/or the push server so that such applications may use the push messaging framework provided by the push client/push server. As a result, when an application is registered and subsequently is the intended recipient of a push message received from a push server, the message is routed to the application based on information in the push message "specifying which mobile application 406 from a plurality of such applications" is the application identifier of the application as included in the message. *Id.*, 21.

515. However, Houghton-Munson does not appear to expressly provide a mechanism to verify the sender's identity or that the destination application is the intended recipient. Without such verification, a POSITA would have understood that the device and/or the destination application can be vulnerable to attacks by unknown third-party senders.

516. Rakic, for instance, recognized the risk of unverified transmissions, and proposed the use of electronic signatures to verify the parties to the transmission. In the context of push environments and push communications of the type disclosed in Houghton and Munson, Rakic provides one well-known solution to address the above-described computer security concerns. As described in §VIII.C (Rakic Overview) and Ground 1D (which I incorporate here), Rakic discloses a database device 108 that stores information related to client devices (e.g., codes obtained from a client device such as IMEI or a component in the client device (e.g., removable memory), and a secure push message server 504 uses stored information in the database device 108 to generate a key, an electronic signature, and a secure push message for a push message intended for a particular client device. Ex-1015, [0047]-[0049], [0065].

517. The secure push message server generates the key based on the above-described code(s) (e.g., IMEI) that are obtained from client device 102 or a component thereof, and subsequently, generate an electronic signature based on the key and the message. *Id.*, [0066]. The secure push message server 504 generates and sends the secure push message to client device 102, e.g., by appending the electronic signature to the message and/or other information (e.g., a timestamp). *Id.*, [0067]. At the client device, a secure push message client can validate the message and the sender by generating a key and then generating a signature using this generated key

and the data block of the received message. *Id.*, [0109]-[0112]. If the signature generated at the push message client matches the signature portion of the received message, the secure push message is determined to be valid. *Id.*

518. In my opinion, a POSITA would have been motivated to implement the above-described techniques in Rakic within Houghton's push server to address the above-described security problems. Indeed, a POSITA would have been motivated to do so because, in view of Rakic, the above-described techniques enable (1) the client device to "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." Ex-1015, [0041]. In doing so, a POSITA would have recognized the improvements to data and computer security for both the client device and the underlying applications to which the data is being transmitted. *Id.*

519. In view of the above motivations, a POSITA would have implemented Rakic's secure push message server 504 and database device 108, as well as their above-described functionality, as part of Houghton's push server. So, configuring the Houghton-Munson's system to implement Rakic's teachings would have amounted to the use of a known technique (electronic signatures) to improve similar devices (computer systems) in the same way (added security through verification), and combining prior art elements (electronic signatures and computer systems)

according to known methods (secure servers, databases, and electronic signatures) to yield predictable results (enhanced verification-based security).

520. Indeed, given that each of Houghton, Munson, and Rakic contemplate push messaging with a similar push architecture involving a push server (e.g., push server and Secure Push Message Server) that communicates with a push client/user device, a POSITA would have found it straightforward—with a reasonable expectation of success—to implement a secure server, including the signature generation functionality of Rakic's secure push messaging server, as well as its database storing the client device information, within Houghton's push server.

521. Additionally, the resulting system's elements would each perform functions they were known to perform prior to the combination—i.e., Houghton-Munson's system would continue to facilitate communication of messages between push server and push client, and Rakic's teachings (as implemented in combination) would facilitate provision of secure signatures (along with the messages) to the message receiving device(s).

522. In the resulting combination, Rakic's secure push message server and its associated signature generation functionality would be implemented in the push server and would be used to generate signatures for messages intended for different mobile terminals and their respective push clients.

523. A POSITA would have understood that the signatures are authorization signatures because they are used for authentication of the sender and/or verify that the intended recipient of the message is the mobile terminal and that the mobile terminal includes a correct component. Ex-1015, [0041].

524. Additionally, as Rakic explains, the signature generating functionality "verif[ies] that an intended recipient of the secure push message is the client device, and that the client device includes a correct component." *See id.*

525. A POSITA would have found it obvious that one such component could be an application of the device. In other words, Rakic does not limit its teachings of verification of component to hardware components, and in the context of push communications that are intended to be delivered to specific software components/applications, a POSITA would have found it obvious that such components would include software components. As such, in view of Rakic as applied in the context of Houghton and Munson, a POSITA would have found it obvious that the above-described secure authorization signatures would indicate the authorized application(s) (i.e., software components) that are allowed/authorized to receive messages including application data via the push server.

M. Ground 2F: Claims 9 and 10 Are Rendered Obvious by Houghton, Munson, and Adamczyk

1. ***Claim 9:*** *The message link server of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

526. In my opinion, if PO contends that claim 9 is not rendered obvious by Houghton-Munson, Houghton-Munson combined with Adamczyk further confirms its obviousness.

527. Adamczyk discloses a message-transmission system like Houghton-Munson. *See, e.g.*, Ex-1021, Abstract, [0007]-[0009]. Adamczyk, however, expressly describes how 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 the preferences of the user. *Id.*, [0010], [0022], cl. 4 ("wherein the notification server is configured to send the notification messages to the recipients on demand, at a specific future time, and/or on a periodic schedule").

528. Although Houghton-Munson discloses a series of triggers (which I describe in (1[d3]/15[d2]) and incorporate here), in my opinion, a POSITA would have appreciated that Houghton-Munson's triggers could be supplemented with the additional types of triggers like a periodic schedule expressly disclosed by Adamczyk. Specifically, a POSITA would be motivated to provide users the option to schedule delivery of messages—including on a "periodic schedule," as taught by Adamczyk, sufficient to control when messages would be pushed to their devices.

529. In my opinion, a POSITA would have been motivated to add the periodic timer trigger to those Houghton-Munson already discloses for several

reasons. First, additional message delivery triggers create a more configurable system that better suits a wider range of user needs in an efficient manner.

530. Second, a POSITA would have appreciated that periodic timers would arguably improve the battery life of the device operating the push client. Rather than have to establish a connection between the push client and push server every time a message is received by the push server, or have the push client repeatedly ping the push server, the number and timing of connections and retrievals could be limited to pre-set intervals. *See, e.g.*, Ex-1040, [0074] (describing battery benefits of setting intervals for call connections rather than persistent connections).

531. In my opinion, a POSITA would have had a reasonable expectation of success in combining these two concepts because the Houghton-Munson combination already taught timers to control the delivery of messages and periodic messaging. Indeed, configuring Houghton-Munson's push server to push messages at predetermined, periodic intervals would simply require configuring the push server to hold received messages until expiry of a recurring timer or periodic scheduled event before pushing all messages received during the time interval at once.

2. ***Claim 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.*

532. Houghton-Munson-Adamczyk renders obvious that the period of the timer is fractionally shorter than a maximum data message interval after which the secure message link is taken down for the same reasons I have explained previously with respect to Houghton-Munson in claim 10 (which I incorporate here), and Houghton-Munson-Adamczyk in claim 9 (which I incorporate here). Specifically, POSITAs would have understood and found obvious that the period of the periodic timer trigger in Adamcyzk would be fractionally shorter than the period for sending the "periodic message" that prevents connection expiration in Houghton-Munson for the same reasons already stated.

N. Ground 2G: Claim 14 Is Rendered Obvious by Houghton, Munson, and Gellens

1. ***Claim 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.*

533. Houghton-Munson-Gellens renders obvious a message delivery trigger based on an amount of wireless network data usage consumed by an end-user device.

534. As I explained previously in 1[c1]/15[c1] and 1[d3]/15[d2] (incorporated here), Houghton-Munson discloses several triggers to control when messages are delivered to a push client from authorized applications. However, Houghton-Munson does not appear to disclose triggering message delivery based on an amount of data consumed by the user terminal.

535. As of the Critical Date (2009), however, wireless bandwidth was not as extensive as it is today and transmission costs per unit data were generally higher. Prior art, such as Gellens, recognized a need to control the delivery of data over cellular networks due to issues such as cost and bandwidth. Ex-1020, Abstract. Specifically, Gellens recognized that applications incorporated into network systems were sending an increased number of messages to end users—many of which were "low priority," yet transmitting these messages could drive up costs to establish the connections to deliver these messages, leading to "excessive and wasteful traffic channel usage." *Id.*, [0004] ("Especially when the amount of data to be exchanged is small, the cost of bringing the traffic channel up and down becomes high compared to the amount of data exchanged. By contrast, when a larger amount of data is exchanged, the cost of bringing the traffic channel up and down is spread over more traffic."); *see also id.*, Abstract, [0002]-[0006], [0027], [0029].

536. To address this problem, Gellens proposes using triggers (like Houghton-Munson) to control when messages are delivered from applications to clients. Gellens provided this control by allowing applications to be assigned priorities, and messages from low-priority applications would be delivered only after a higher-priority application initiated a link and transmitted its message. *Id.*, [0035]-[0036]. That way, connections would not be established and maintained for *every* message—no matter how minor—but low-priority messages would be queued until

a message of significance opened the connection. In other words, the "trigger' would be the high-priority application using "an amount of wireless network data," as claimed, which would then trigger delivery of the low-priority messages.

537. In my opinion, a POSITA would have been motivated to combine Houghton-Munson and Gellens to prioritize certain applications, and only permit transmission of low-priority application messages after a high-priority application opens a connection and transmits its message, thereby limiting connection/transmission costs, and "the cost of bringing the traffic channel up and down is spread over more traffic." *Id.*, [0004], [0035]-[0036]. In my opinion, this combination would have amounted to applying a known technique (application prioritization) to a prior-art device (a messaging network) to solve a known problem in the art (networking costs) and achieve a predictable result (triggering low-priority message delivery only after a high-priority application establishes a connection).

538. In my opinion, a POSITA would have also had a reasonable success in making this combination because it merely involves configuring the applications Houghton-Munson already envisions being incorporated into the messaging network to be assigned priorities, and control what priorities are required to establish a connection between the push server and push client.

X. ANALYSIS OF THE '320 PATENT

A. Ground 1A: Claims 1-3, 5-7, 10-11, 15, and 18 Are Rendered Obvious by TS-23.140

1. Claim 1

a. 1[pre]-1[a]: A networked system comprising: a network server system including

539. If the preamble is limiting, in my opinion, TS-23.140 discloses or renders obvious a networked system (e.g., MMS network) comprising a network server system (e.g., MMS Relay/Server).

540. TS-23.140 discloses a network of servers, MMS User Agents, databases, and applications (networked system) that includes an MMS Relay/Server to facilitate delivery of messages between different entities over a network (as shown below):

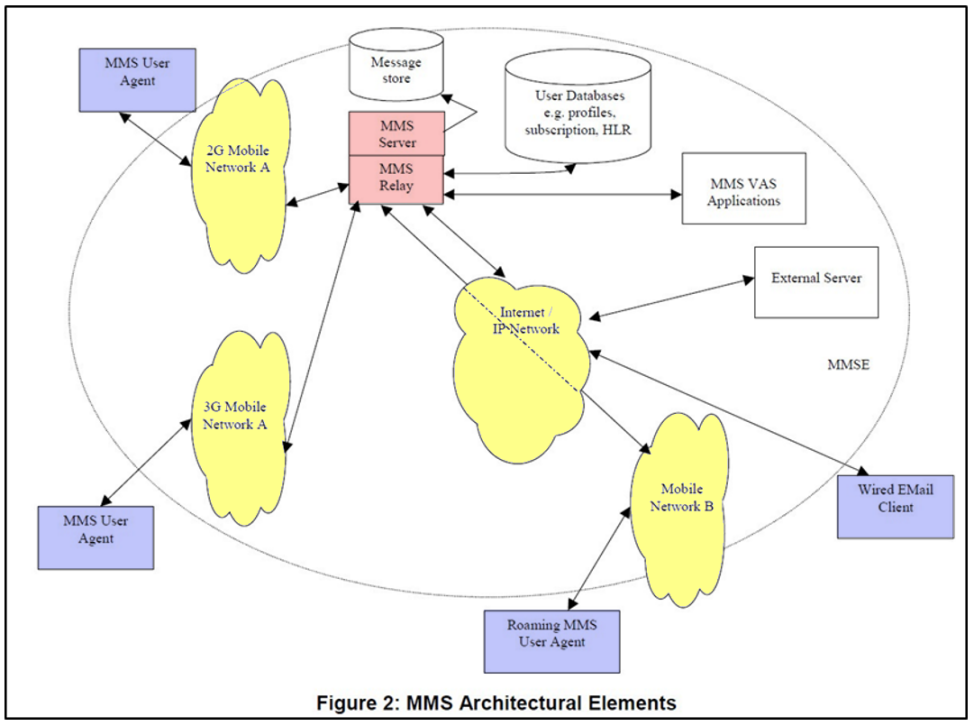


Figure 2: MMS Architectural Elements

Ex-1004, Fig. 2

541. The MMS Relay/Server "is responsible for storage and handling of incoming and outgoing messages and for the transfer of messages between different messaging systems." *Id.*, 17 (noting that this server may be a single logical element or may be separated into MMS Relay and MMS Server elements; describing generation of charging data records (CDR) when retrieving or delivering multimedia messages (MM) and for Value Added Service Provider (VASP) operations).

542. As shown above, MMS communications are facilitated between MMS User Agents (i.e., between two user equipment (UEs)) or between MMS User Agents and MMS VAS Applications.

543. The MMS User Agent facilitates transmission and receipt of multimedia messages (MMs) between different devices (e.g., MMS User Agents executing on other UEs, MMS Relay/Server, MMS VAS Applications) that are linked together over a communication network (as shown in Figure 2 above). *Id.*, 19-20, §5.1.1 (describing operations of MMS User Agent). The MMS User Agent is an application residing on a UE, whereas an MMS VAS Application is an application that provides Value Added Services (e.g., news service or weather forecasts) to MMS Users and resides on a separate network entity, as shown in the above figure. *Id.*, 14.

544. MMS can also be "used to transport data specific to applications." *Id.*,54. Such applications can be downloaded "to a mobile phone" and can register

with an MMS User Agent of the UE (user equipment) or an MMS VAS application. *Id.* Once registered, MMS can be used "to transport 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.

545. These various elements described in TS-23.140 constitute a networked system, with the MMS Relay/Server being a network server system.

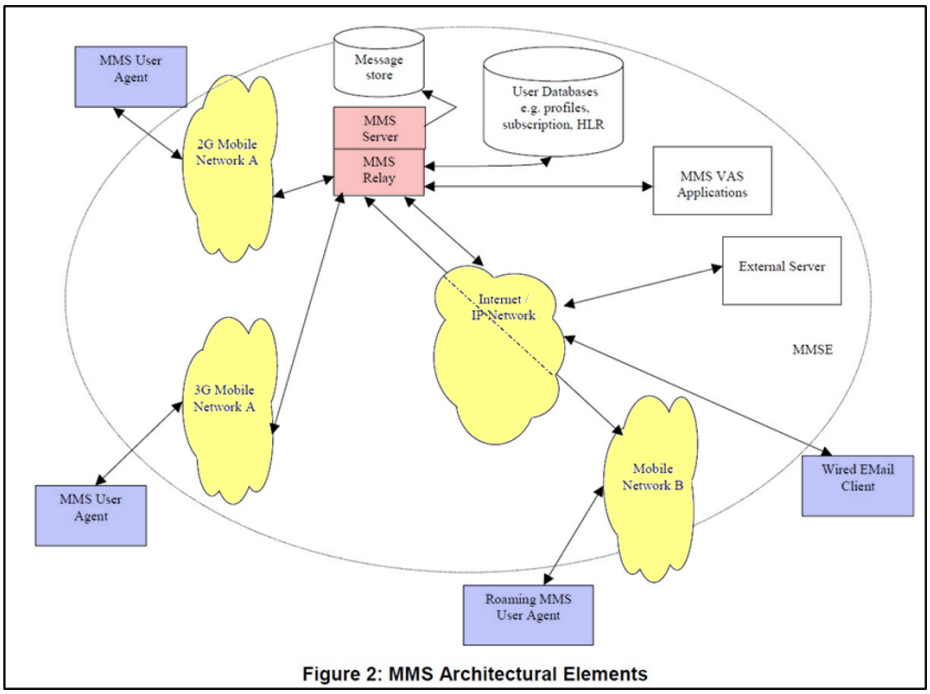
546. Additional details regarding the MMS Relay/Server and the method for operating it are provided below with reference to 1[b1]-1 [e4] *infra*.

547. For the above reasons, TS-23.140 renders obvious 1[pre]-1[a].

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

548. In my opinion, TS-23.140 discloses or renders obvious a link interface (comprising the MM1 Transport Protocol and TCP/IP and transport layer security (TLS)-based transport protocols) to maintain a respective secure message link (e.g., secure links) for sending data through an Internet network between the link interface of the network server system (e.g., MMS Relay/Server), to a respective device link agent (e.g., MMS User Agent) on one of a plurality of wireless end-user devices (e.g., wireless client devices).

549. In more detail, and as shown in Figure 2 (reproduced below), the MMS Relay/Server communicates via an Internet network with MMS User Agents that execute on each UE (wireless end-user device).



Ex-1004, Fig. 2.

Id., 17 ("Figure 2 shows that multimedia messaging may encompass many different network types … provided by the Internet protocol and its associated set of messaging protocols" that "enables messaging in 2G and 3G wireless networks to be compatible with messaging systems found on the Internet."), 19-20 (describing MMS User Agent facilitating transmissions over a network).

550. Similarly, as disclosed in Ex-1010, Figure 3 depicts a similar MMS communication between an MMS client and the MMS Relay/Server over the Internet and a wireless network.

551. The MMS User Agent is a device link agent because it facilitates transmission and receipt of multimedia messages (MMs) between different devices (e.g., MMS User Agents executing on other UEs, MMS Relay/Server, MMS VAS Applications) that are linked together over a communication network (as shown in Figure 2 above). *Id.*, 19-20, §5.1.1 (describing operations of MMS User Agent); *see also* Ex-1036, 2, Fig. 1 (showing multiple MMS User Agents in communication with MMS Relay/Servers).

552. The '320 patent does not specifically describe the structure of the link interface (which appears only in the claims), and instead claims its functionality. However, I also understand that the familial '192 Patent uses the term "transport services stack" to perform several of the same claimed functions as the "link interface" claimed in the '320 Patent. I have provided a table below showing how these terms appear to be used in similar contexts:

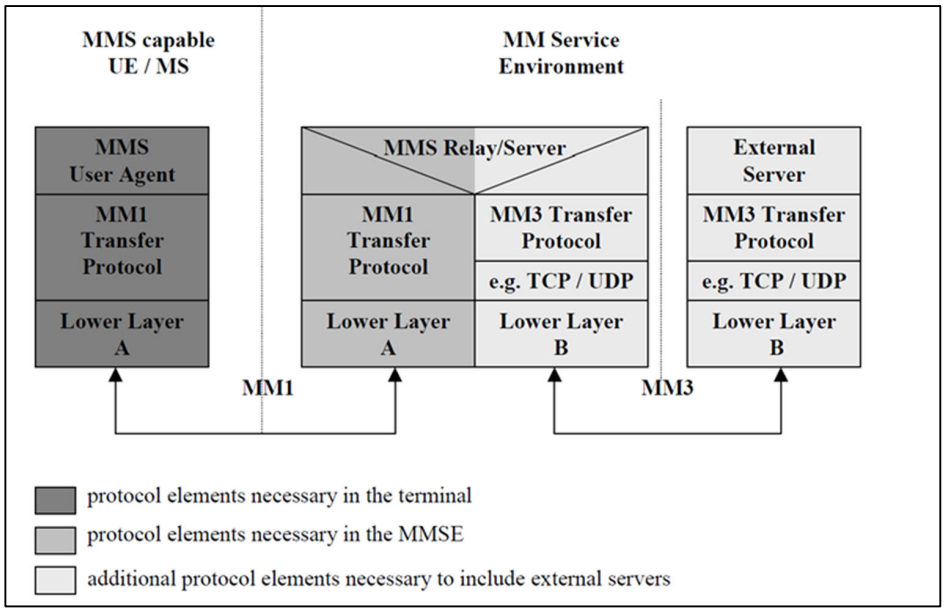
'192 Patent	**'320 Patent**
1[a]. ***a transport services stack*** to maintain a respective secure message link through an Internet network between the message link server and a respective device link agent on each of a plurality of wireless end-user devices	1[b1]. ***a link interface*** to maintain a respective secure Internet data message link between the link interface and a respective device link agent on each of a plurality of wireless end-user devices

7. initiate the respective secure Internet data message link ***to the transport services stack***	1[e2]. maintain the respective secure Internet data message link over a wireless network ***to the link interface***

The ’320 specification further states that a “transport services stack” facilitates communications between the client and server devices and can be “implemented with standard secure or open Internet networking protocols, such as TLS or TCP.” *See* Ex-1001, (’320 Patent), 89:24-41, 90:34-50. In other words, I understand the claimed “link interface” to at least include a stack of transport protocols including (among others) TLS and TCP.

553. A POSITA would have also understood TLS to refer to Transport Layer Security, and TCP as referring to Transmission Control Protocol. *See* Ex-1027, Ex-1028.

554. Consistent with this disclosure, TS-23.140 also discloses or renders obvious a link interface in the form of transport protocols (e.g., MM1) and associated functionality (that uses TCP and TLS) that maintains a secure message link through an Internet Network between the message link server (e.g., MMS Relay/Server) and wireless end-user devices (e.g., UEs using the same protocol). For example, TS-23.140 discloses that MMS shall “make use of the protocol framework outlined in figure 4” pursuant to which the MMS User Agent communicates with MMS Relay/Server. Ex-1004, §6.2, pp. 24-25.



Id., §6.2, pp. 24-25, Fig. 4 (providing protocol framework to provide MMS; listing MM1 transfer protocol as part of MMS Relay/Server).

555. As shown, TS-23.140 discloses that the MM1 Transfer Protocol is implemented both at the MMS Relay/Server and the MMS UE. *See id.* A POSITA would have understood that the MM1 Transfer Protocol is a link interface that facilitates transmission/transport of network communications between the MMS Relay/Server and network elements (e.g., the MMS User Agent, MMS VAS Applications).

556. A POSITA would have understood that TS.23.140 discloses a "link interface" (as claimed) comprising the MM1 Transfer Protocol with TCP/IP and TLS protocols given its well-known description in the art as the MM1 Transfer

protocol that provides a "transport mechanism" using the HTTP/TCP/IP. Ex-1013, 11.

557. Indeed, consistent with this understanding, TS-23.140 contemplates "WAP/OMA implementation" for the "MM1 Transfer Protocol" and incorporates by reference Open Mobile Alliance specifications, such as those provided in Ex-1011. *See* Ex-1004, 13, 162. This incorporated document explains that "[s]ince MMS is an application-level protocol framework to enable messaging transactions, a device implementing OMA MMS must have either of WAP WSP stack or HTTP/TCP/IP stack." Ex-1011, 11.

558. Further still, the same suite of OMA specifications related to MMS explains that "TLS" or "transport layer security protocol provides for secure data transmission between the MMS Client and the MMS Proxy-Relay in architectural configurations that employ HTTP based protocol stacks for MMS_M implementation." Ex-1010, 22.

559. Given these disclosures and the well-known use of TLS to secure/encrypt network communications, a POSITA would have understood or found it obvious that the link interface in the MMS Relay/Server and MMS User Agent would use TLS for securing the communication link between the MMS User Agent and the MMS Relay/Server. This is further corroborated by Ex-1010, which contemplates "encrypt[ing] the communication channel between the client and the

backend MMS server with some security protocols, e.g., WTLS (Wireless Transport Layer Security).” Ex-1014, [0017].

560. A POSITA would have further understood or found it obvious that in the MMS environment, multiple MMS User Agents (and their associated UEs) are in communication with the MMS Relay/Server, such that a separate secure TLS-based link is maintained between each MMS User Agent and the MMS Relay/Server. This is further corroborated by multiple references. As a starting point, a POSITA would have known that, in the relevant timeframe, TLS or SSL-based communications are expensive to form and use keep-alive and session-resumption methods to maintain already-formed/established connections (and thereby avoid cost and time to re-establish such connections). Indeed, this is corroborated by Houghton, which discloses connections between push servers, akin to MMS Relay/Server, and push clients, akin to MMS User Agents, being “persistent managed, tested and configured” and use periodic messages to prevent “time expir[ing]” the connection. Ex-1007, 23, 17. For example, Mostafa discloses and shows (as reproduced below) communication links formed and maintained between MMS Relay/Server and multiple terminals (and their respective MMS User Agents):

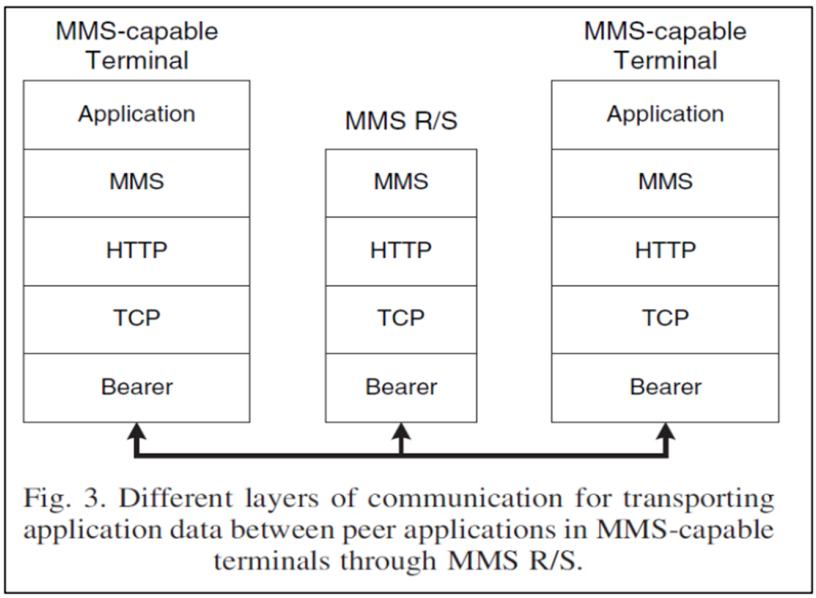


Fig. 3. Different layers of communication for transporting application data between peer applications in MMS-capable terminals through MMS R/S.

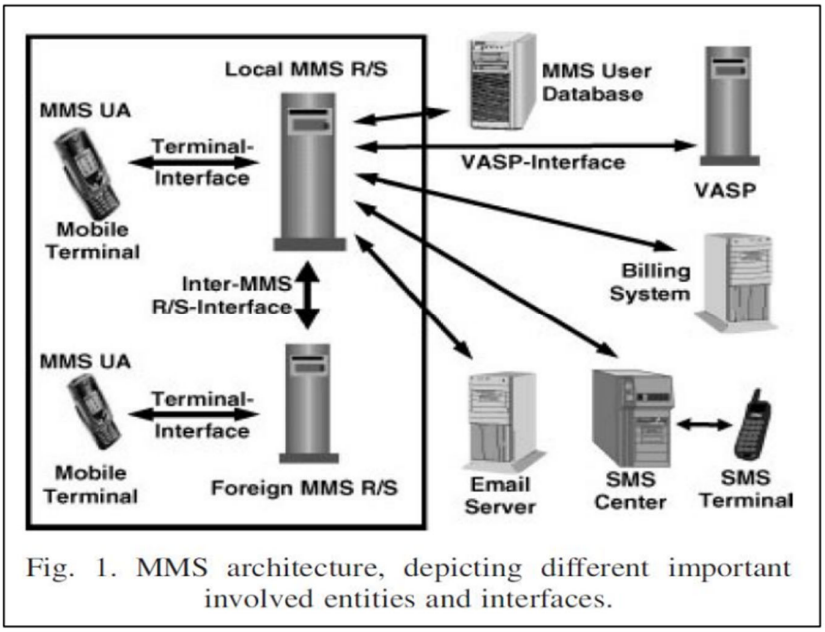


Fig. 1. MMS architecture, depicting different important involved entities and interfaces.

Ex-1036, Figs. 1-3, pp. 2-4. Munson further corroborates that such persistently maintained communication links are formed between multiple devices and the servers, as shown below.

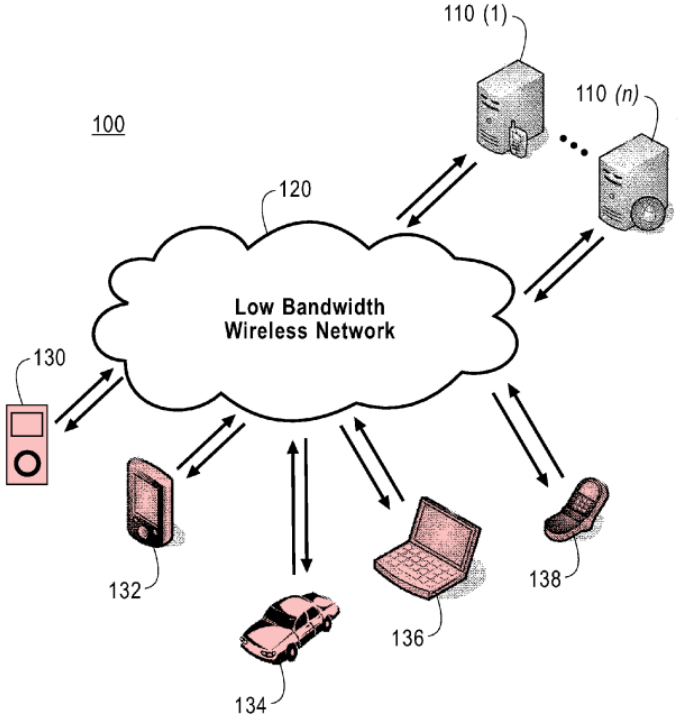


FIG. 1
Prior Art

Ex-1017, Fig. 1, [0007]-[0008] (showing plurality of data connections with multiple end-user devices).

561. For the above reasons, TS-23.140 discloses or renders obvious 1[b1].

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

562. TS-23.140 discloses or renders obvious that each of the wireless end-user devices (e.g., wireless client devices) include multiple software components (e.g., multiple applications) authorized to receive messages via a device link agent (e.g., MMS User Agent) on that device.

563. In more detail, TS-23.140 discloses that applications, such as those that are downloaded on a mobile phone, can use MMS to "transport data specific to ***applications***" and MMS communication can include "transport data specific to applications between two MMS User Agents or an MMS User Agent and an MMS VAS Application." Ex-1004, §7.1.18, pp. 54-55. Thus, TS-23.140 discloses that each UE (wireless end-user device) includes multiple software applications (software components). Ex-1036, 2-4 (showing and listing multiple applications in UEs that receive application data via MMS), Fig. 5 below.

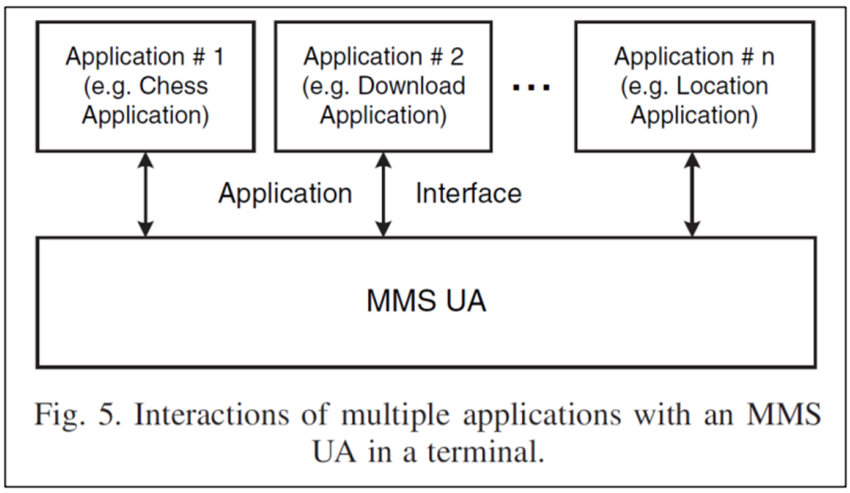


Fig. 5. Interactions of multiple applications with an MMS UA in a terminal.

564. A POSITA would have understood or found it obvious that each such application has to be authorized to receive and process messages via MMS. TS-23.140 discloses that "[a]pplications that want to transport data specific to applications other than MMS will initially ***need to register with the appropriate MMS User Agent or MMS VAS Application***." Ex-1004, §7.1.18, pp. 54-55. "During this registration process the application provisions an MMS User Agent or an MMS

VAS Application with its application identification value and may negotiate with the MMS User Agent or MMS VAS Application the details (amount and format) of information to be exchanged between the two entities." *Id.*

565. Additionally, the MMS User Agent can indicate the "Terminal Capability" toward its recipient MMS Relay/Server by transmitting "a set of information describing the terminal's capabilities, including a binary flag indicating whether or not the terminal supports Application data." *Id.*, §7.1.3.1, p. 30.

566. Subsequently, a message including application data for an intended application is only delivered to that application upon determining that the MMS User Agent has the capability of supporting application data and the terminal includes the intended application for the receipt of such application data. *Id.*, pp. 30, 54-56 ("If the MMS Relay/Server finds from the recipient MMS User Agent's capability indication … that the recipient MMS User Agent supports transport of application data, the MMS Relay/Server … shall pass on the destination application identifier, the 'reply-path' identifier (if present) and the additional application/implementation specific control information (if present) unaltered."; "If the destination application does not reside on the receiving MMS User Agent or on the USIM or MMS VAS Application, the MMS User Agent or MMS VAS Application shall discard the corresponding abstract message."). From these disclosures a POSITA would have

understood that such applications are ***authorized*** to receive messages including application data.

567. A POSITA would have further understood that application data received by a particular destination application would be processed by that application to, e.g., facilitate display of that data or perform operations with respect to that data. Ex-1004, § 7.1.18.2.2, 56 ("rout[ing] the received MMS information on to the destination application" and contemplating "further handling and processing of the" received application data "by the destination application").

568. Further still, because the above-described communication is facilitated between MMS User Agents and/or an MMS User Agent and an MMS VAS Application, and such communications occur via the MMS Relay/Server (as described above in 1[b1]), a POSITA would have understood that the messages including the application data are received via the MMS User Agent (device link agent) on the particular terminal/device. *Id.*, §7.1.18, pp. 54-55 ("Based on the negotiated details upon application registration process an application may trigger an MMS User Agent or an MMS VAS Application to submit certain abstract messages. Upon triggering an MMS User Agent or an MMS VAS Application to send an abstract message the MMS User Agent or MMS VAS Application may receive information from the application. ***The MMS User Agent or MMS VAS Application may insert this information in both the information elements and/or***

payload (if present) of the abstract message…. Abstract messages that are sent by an MMS User Agent or an MMS VAS Application on behalf of an originating application shall contain a destination application identifier.”); *id.*, 56 (“Upon reception of an abstract message containing a destination application identifier (it can either be the MM1_notification.REQ, MM1_retrieve.RES or MM7_deliver.REQ transactions), ***the receiving MMS User Agent or MMS VAS*** *Application shall first check if the destination application resides on it*” and if so, the *MMS User Agent shall immediately route the received MMS information on to the destination application that is referred to from the destination application identifier*”).

569. As I explained above in 1[b1], TLS provides for secure data transmission between the MMS Client and the MMS Proxy-Relay in architectural configurations that employ HTTP based protocol stacks for MMSM implementation.” Ex-1010, 22. Therefore, in my opinion, once registered, application-specific MMs may be delivered by the Relay/Server using a protocol that includes TLS to the registered application.

570. A POSITA would have found it obvious, in view of TS-23.140’s disclosures, that the above-described procedure of delivery of application-specific data to an application would be performed for multiple applications that are intended to send or receive application-specific data. For example, TS-23.140 provides

examples of applications (e.g., chess application, messaging/discussion applications, news applications, weather applications) that use MMS for application specific messaging. *See, e.g.*, *id.*, p. 55 ("The additional application/implementation specific control information shall be used for all future needs that are not supported by the application identifier of the destination application and the identifier of the originating application, such as specifying a particular logical channel in the application addressing method (e.g., 'discussion thread #05') or distinguishing between multiple instances of the same application (e.g., 'chess application #02'); p. 14 (explaining that VAS application are "applications providing value added services (e.g. news service or weather forecasts) to MMS users"), pp. 54-55 ("Applications that … transport application specific data using MMS … reside on an MMS User Agent…. Applications that want to transport data specific to applications other than MMS will initially need to register with the appropriate MMS User Agent"; describing different types of applications, e.g., chess and messaging applications"). Indeed, MMS provided a well-established infrastructure for application messaging and as described above, applications would naturally use MMS messaging to facilitate mobile content delivery (e.g., delivery of news, weather, or application related messaging).

571. Further still, by 2009, it was well-known for mobile/MMS-capable user devices/terminals to include multiple applications and for such applications to send

and receive application-specific data using MMS. For example, Mostafa discloses and depicts (below) multiple applications, e.g., chess, download, and location applications, that interface with the MMS User Agent (UA) to facilitate sending and receiving of application data via MMS. *See* Ex-1036, 3-4 (pp. 732-33, Figs. 3-5).

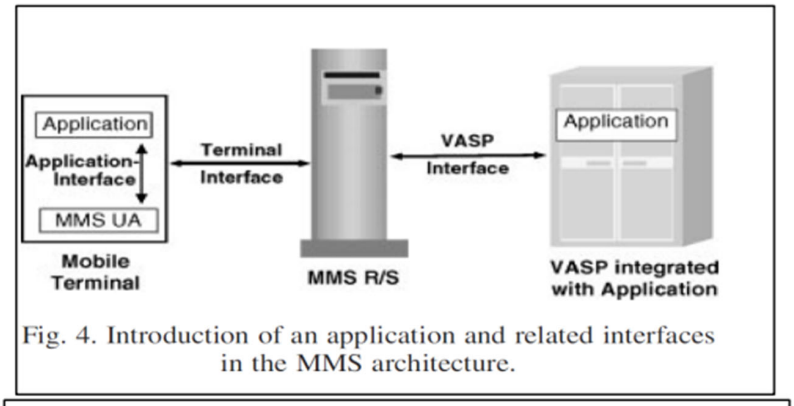


Fig. 4. Introduction of an application and related interfaces in the MMS architecture.

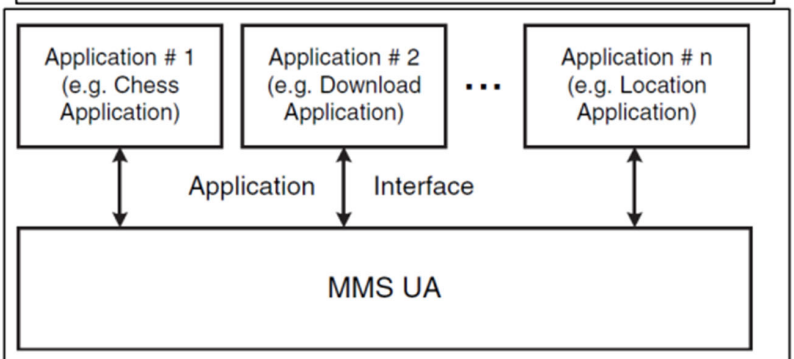


Fig. 5. Interactions of multiple applications with an MMS UA in a terminal.

See also id., 5 ("As for example, a chess application within a terminal may offer its user to play chess with a user of a different terminal having the same application. If both the terminals are also MMS-capable, the application may use an MM to convey the information about each move made by a user on the chessboard to the other user. Other interactive games (e.g., golf, snooker) in a mobile terminal can also use MMS

as the transport similarly. A location service could also be based on this new MMS feature, so that a location server, acting as an application over a VASP, can provide location-specific information to a terminal. Moreover, the feature is also useful for downloading any content (e.g., a media object, a new game, a new plug-in of an existing game, an application, media codec) from a VASP. Furthermore, a configuration application could use the feature to get a new or an updated configuration of a terminal (e.g., connectivity information for any network service like MMS, browsing) from a network. These are just few examples, and the feature can be used for many other purposes.").

572. Thus, a POSITA would have understood or found it obvious that the UEs (i.e., the wireless end-user devices) would include multiple applications (i.e., software components) that are capable of receiving and processing the registered application data received from the MMS Server/Relay.

573. For the above reasons, TS-23.140 discloses or renders obvious 1[b2].

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

574. TS-23.140 discloses or renders obvious a network interface (e.g., multiple interfaces including MM1, MM3, MM7) to receive messages (e.g., messages including application-specific data and other addressing data) from a plurality of network elements (e.g., MMS User Agents resident on MMS client

devices, MMS VAS Applications) for delivery to the specific application identified in the message.

575. In more detail, the following figure shows an MMS network environment in which communications are facilitated between various MMS User Agents, the MMS Server/Relay, MMS VAS Applications, and one or more external servers over various networks.

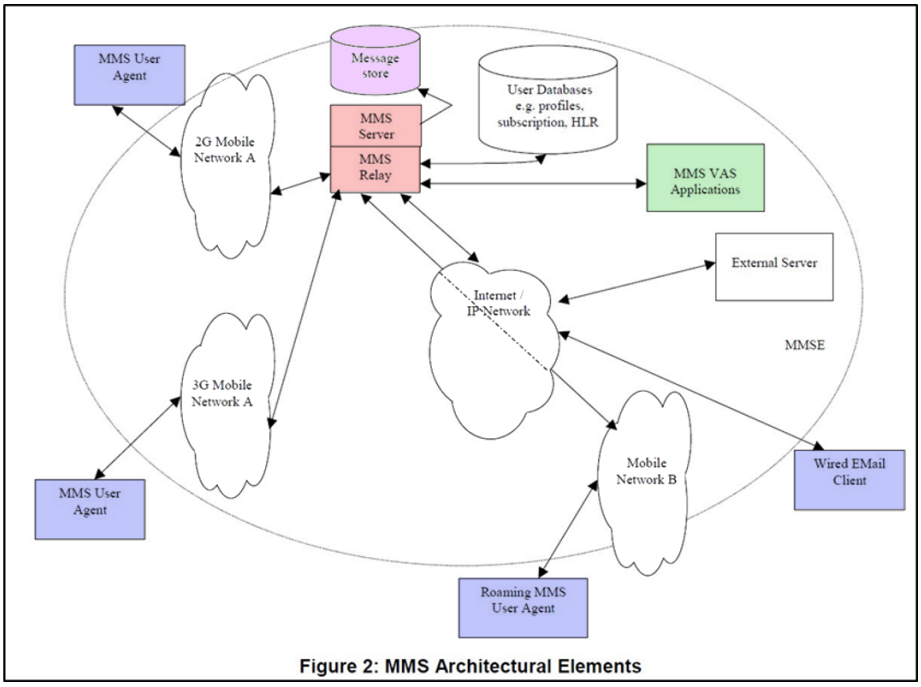


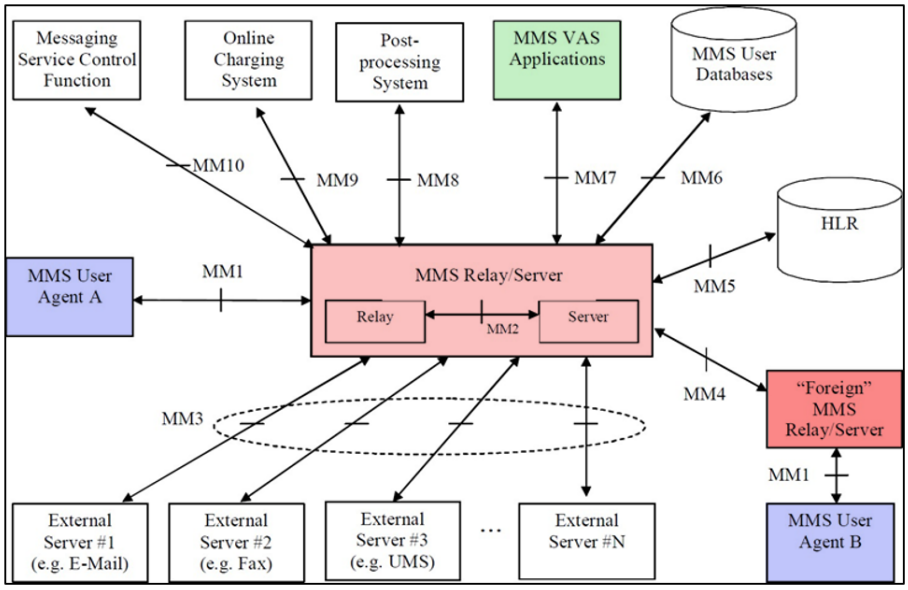
Figure 2: MMS Architectural Elements

Ex-1004, § 4.2, Fig. 2, p. 17 (describing the above MMS environment as "***a collection of MMS-specific network elements***.").

576. As described with reference to 1[b1]-1[b2], the MMS Relay/Server receives messages from one or more of these network elements (i.e., network

element messages), such as one or more MMS User Agents and MMS VAS Applications.

577. TS-23.140 further discloses various "interfaces" facilitating network communication of such network element messages. *Id.*, §6.1, pp. 23-24. For example, the following figure shows the various interfaces facilitating communication between MMS User Agents, MMS VAS Applications, and MMS Relay/Server, including the MM1, MM2, MM7, and MM4 interfaces:



Id., §6.1, Fig. 3, pp. 23-24.

578. Specifically, TS-23.140 discloses that (1) the MM1 interface is the "reference point between the MMS User Agent and the MMS Relay/Server," (2) the MM4 interface is the "reference point between the MMS Relay/Server and another MMS Relay/Server that is within another MMSE," and (3) MM7 interface is "the

reference point between the MMS Relay/Server and MMS VAS Applications." *Id.*, §6.1, pp. 23-24.

579. As shown in the above Figures, the interfaces are to various networks, including 2G/3G mobile networks, and IP/internet networks. *Id.*, Fig. 2.

580. Additionally, as described with reference to 1[c2] below, applications that are registered with MMS User Agents or MMS VAS application can transmit application data to applications registered with other MMS User Agents and VAS applications, in which case the communications are facilitated via the MMS Server/Relay. *See* Ex-1004, §7.1.18, pp. 54-55 (contemplating delivery of application specification data to an application via the user agent, using that application's identifier). As I explain in further detail in 1[e3], application MMs include both application data and the "application identifier of the destination application." *Id.*, 54-55. The originator also "indicate[s]" the message recipient's address in the message. *Id.*, 26, 90, 190 (including "Recipient address" in messages sent over MM1). MMS Relay/Server receives this message and passes "on the destination application identifier" and "application data" to, e.g., another MMS User Agent. *See id.*, 54-56. In other words, application messages being delivered identify the application to which they are directed.

581. For the above reasons, TS-23.140 discloses or renders obvious 1[c1].

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

582. TS-23.140 discloses or renders obvious that each network element (e.g., MMS User Agents resident on MMS client devices, MMS VAS Applications) authorized to send messages via the link interface (comprising the MM1 Transport Protocol and TCP/IP and transport layer security (TLS)-based transport protocols) to software components (e.g., applications) on a user wireless device.

583. TS-23.140 discloses that, to transmit over the MM1 protocol or another network interface protocol as discussed above in 1[c1], the application must register with the MMS User Agent, which then sends the transmitted message to its destination. *Id.*, 54-56. These registered applications utilize an MM1 Transfer Protocol with TCP/IP and TLS security to communicate. As I noted above, a POSITA would have understood that TS.23.140 discloses that the MM1 Transfer Protocol utilizes TCP/IP and TLS protocols given its well-known description in the art as the MM1 Transfer protocol that provides a "transport mechanism" using the HTTP/TCP/IP. Ex-1013, 11.

584. The destination application is also a registered application that has undergone the same registration process with the requisite MMS User Agent. *Id.*, 54-56. A POSITA would have understood that sending and receiving in the TS-23.140 system would require registration since the applications are registering with

the MMS User Agent to be able to use the secure communication channels and ensure the security of the transmission on both ends.

585. In more detail, TS-23.140 discloses that applications, such as those that are downloaded on a mobile phone, can use MMS to "transport data specific to ***applications***" and MMS communication can include "transport data specific to applications between two MMS User Agents or an MMS User Agent and an MMS VAS Application." *Id.*, §7.1.18, pp. 54-55. Thus, a POSITA would have understood that TS-23.140 discloses that each UE (wireless end-user device) may include multiple software applications (software components). *Id.*; Ex-1036, 2-4 (showing and listing multiple applications in UEs that receive application data via MMS), Fig. 5 below.

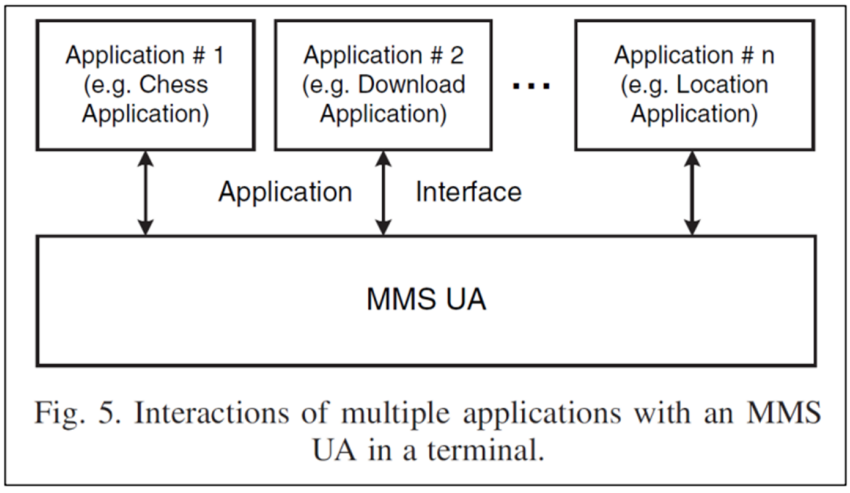


Fig. 5. Interactions of multiple applications with an MMS UA in a terminal.

586. A POSITA would have understood or found it obvious that each application must be authorized to send messages via MMS. TS-23.140 discloses that "[a]pplications that want to transport data specific to applications other than MMS

will initially ***need to register with the appropriate MMS User Agent or MMS VAS Application***." Ex-1004, §7.1.18, pp. 54-56. "During this registration process the application provisions an MMS User Agent or an MMS VAS Application with its application identification value and may negotiate with the MMS User Agent or MMS VAS Application the details (amount and format) of information to be exchanged between the two entities." *Id.*

587. Additionally, the MMS User Agent can indicate the "Terminal Capability" toward its recipient MMS Relay/Server by transmitting "a set of information describing the terminal's capabilities, including a binary flag indicating whether or not the terminal supports Application data." Ex-1004, §7.1.3.1, p. 30.

588. Subsequently, a transmitted message including application data for an intended application is delivered to that application only upon determining that the MMS User Agent has the capability of supporting application data and the terminal includes the intended application for the receipt of such application data. *Id.*, pp. 30, 54-56 ("If the MMS Relay/Server finds from the recipient MMS User Agent's capability indication … that the recipient MMS User Agent supports transport of application data, the MMS Relay/Server … shall pass on the destination application identifier, the 'reply-path' identifier (if present) and the additional application/implementation specific control information (if present) unaltered."; "If the destination application does not reside on the receiving MMS User Agent or on

the USIM or MMS VAS Application, the MMS User Agent or MMS VAS Application shall discard the corresponding abstract message."). From these disclosures a POSITA would have understood that such applications are ***authorized*** to send messages including application data to specific applications on other devices.

589. For the above reasons, TS-23.140 discloses or renders obvious 1[c2].

f. ***1[d1]-[d2]****: a message buffer system including a memory and logic, the memory to buffer content from the received network element messages for which delivery is requested to;*

590. TS-23.140 discloses or renders obvious a message buffer system including a memory that buffers content from the received network element messages (e.g., message store for storing the received message from, e.g., an application associated with the originating MMS User Agent or VAS application, for which delivery is requested to a given one of the wireless end-user devices (e.g., an application running on a client device associated with the MMS User Agent for that device).

591. As described in 1[c1]-1[c2] (which I incorporate here), the network element messages are received from one or more network elements by the MMS Relay/Server, and delivery of these messages is requested to an application resident on a user device (and associated with its MMS User Agent).

592. Moreover, the "***MMS Relay/Server is responsible for storage*** and handling ***of incoming and outgoing messages*** and for the transfer of messages

between different messaging systems." Ex-1004, § 4.2, p. 17. Fig. 2 of TS.123.140 shows a message store coupled to the MMS Server/Relay:

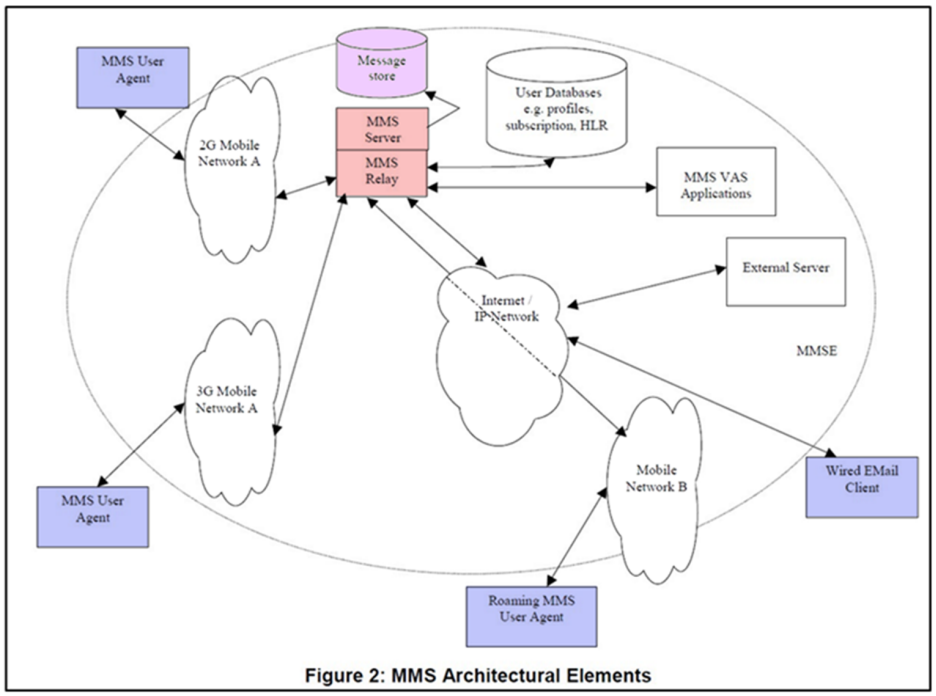


Figure 2: MMS Architectural Elements

See also id., § 5.2, p. 21 (describing "temporary storage of messages" received and send by MMS User Agents; facilitating transport of application data).

593. "Upon reception of an MM from an originator MMS User Agent," the "originator MMS Relay/Server . . . ***shall retain the MM until the earliest desired time of delivery***, if the optional feature of earliest time of delivery is supported by the originator MMS Relay/Server. . . ." *Id.*, § 7.1.1, 7.1.2, pp. 26-28.

594. Similarly, "***[u]pon reception of an MM the recipient MMS Relay/Server*** [which can be the same as the originator MMS Relay/Server]," this MMS Relay/Server can (1) ***store the MM at least until one of*** "the associated time of expiry is reached, the MM is delivered, the recipient MMS User Agent requests

the MM to be routed forward or the MM is rejected," and (2) "shall generate a notification to the recipient MMS User Agent." *Id.*

595. TS-23.140 further describes "persistent storage of messages" in a "Persistent Network-Based Storage" (called MMBox) ***which is associated with "the MMS Relay/Server and into which Multimedia Messages (MMs) may be stored, retrieved, and deleted***." Ex-1004, § 5.2.1, pp. 21-22. As such, when a multimedia message is received from an MMS User Agent, the MMS Relay/Server "***may store the MM into an MMBox***" prior to delivery to another network element such as a UE/wireless device. *Id.*, pp. 26-28.

596. Additionally, as described with reference to 1[d3]-1[d4] (which I incorporate here), the MMS Relay/Server includes a message buffer system with logic that facilitates delivery of messages upon occurrence of one or more message delivery triggers.

597. For the above reasons, TS-23.140 discloses or renders obvious 1[d1]-1[d2].

g. ***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;*

598. As explained below, TS-23.140 discloses or renders obvious the message buffer system including logic that determines when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred (e.g., user request for a message, expiration of a message expiry period, determination of user settings or recommendations specifying retrieval mode to use for received message, conformity of message to MMS User Agent specification of a size restriction).

599. Additionally, and as further explained below, TS-23.140 renders obvious that the receipt of a message is not a message delivery trigger and one of the message delivery triggers is an occurrence of an asynchronous event with time-critical messaging needs (e.g., a user request for a received message).

600. In more detail, TS-23.140 discloses that messages are delivered from the MMS Relay/Server to the MMS User Agent when the MMS Relay/Server sends a notification, and the MMS User Agent responds with a retrieval request for delivery. *See, e.g.*, Ex-1004, 66-69.

Table 7: abstract messages for notification of MM in MMS

Abstract message	Type	Direction
MM1_notification.REQ	Request	MMS Relay/Server -> MMS UA
MM1_notification.RES	Response	MMS UA -> MMS Relay/Server

However, TS-23.140 also specifies several conditions to trigger the MMS User Agent to issue the retrieval request and secure delivery.

601. If the claimed "trigger" must be a condition after which the next step is an attempt to deliver the message (e.g., the last traffic light before a destination), TS-23.140 discloses:

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

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

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

If, however, a "trigger" can include conditions that starts or restarts an attempted delivery (e.g., any traffic light on the way to a destination), in addition to the example triggers (1) and (3) I identify above, the following are also triggers:

(4) When the MMS 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 (§7.1.1).

(5) When the MMS 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 MMS Relay/Server receives a retrieval request from the recipient User Agent with a request for deferred delivery, the expiration of a timer or event occurrence at the delivery time (trigger) initiates a delivery attempt from the recipient User Agent and determines that a specified deferred-delivery period is met or event occurrence at the delivery time, triggering an attempted delivery. *See id.*, 28-29 (§7.1.2.1).

But if a "trigger" can include conditions imposed on the messages controlling whether a message will be delivered at all (e.g., any traffic lights ***plus*** road

restrictions, like maximum height/weight to get to a destination), in addition to (1)-(6), the following is another trigger:

(7) When the MMS Relay/Server receives a retrieval request from the recipient User Agent with a "size restriction," a determination that the message conforms to that restriction (trigger) initiates a delivery attempt. *Id*., 29-30 (§7.1.3).

602. Under any possible interpretation, because the MMS Relay/Server must monitor when these events occur or conditions are satisfied, in my opinion, a POSITA would have understood or found it obvious that the MMS Relay/Server includes logic configured to determine when they have occurred for a given UE/mobile device.

603. Additionally, under any possible interpretation, in my opinion, a POSITA would have understood or found it obvious that, for at least some received network element messages, message receipt by the memory store/MMBox alone is not a trigger. Specifically, if a trigger is a condition that controls whether an attempted message delivery is made (conditions (1)-(7)) or is a condition that starts/restarts an attempted delivery (the conditions in (1), (3)-(6)), the receipt of an message without an earliest desired time of delivery (condition (4)) is a trigger. However, the receipt of message with a specified time of delivery (at least some of the received network element messages) is not itself a trigger; attempted delivery is

only triggered by a determination that the earliest desired time of delivery condition is met (conditions (5) or (6)). And if a trigger is a condition after which the immediate next step is an attempt to deliver the message (conditions (1)-(3)), the receipt of the MM—with or without a desired time of delivery—is not a trigger. Under this interpretation, none of the received messages are triggers, which satisfies the requirement that at least some of the received network element messages are not triggers.

604. Further, under any possible interpretation, a POSITA would have understood or found it obvious that at least some of the triggers occur upon an asynchronous event with time-critical messaging needs. For instance, unless a trigger must be a condition after which the immediate next step is an attempt to deliver the message, the receipt of the message in (4) triggers message delivery immediately (i.e., an asynchronous event with time-critical messaging needs). And under any possible interpretation, the manual retrieval in (1) is a trigger occurring upon an asynchronous event with time-critical messaging needs because the user must manually cause the MMS User Agent to request delivery. Ex-1014, 20. The ’320 Patent itself discloses that a manual “user request” for delivery constitutes such a message-delivery trigger. *See* Ex-1001 (’320 Patent), 38:50-63.

605. In my opinion the “triggers” I have identified above are consistent with the ’320 Patent’s description of what constitutes a trigger. For instance, the ’320

Patent explains: "the service control server link 1638 can determine when to transmit based potentially on several parameters, such as one or more of: periodic timer trigger, waiting until a certain amount of service usage or traffic usage has occurred, responding to a service agent message, responding to a service agent request, initiated by one or more servers, initiated by a verification error condition, and/or initiated by some other error condition." Ex-1001 ('320 Patent), 70:6-13. The '320 Patent further discloses: "In addition, the service control server link 1638 can provide for an efficient communication link based on various embodiments related to the timing of transmissions over the service control link, as similarly discussed above with respect to the service control device link 1691 description. For example, the timing functions, such as asynchronous messages or polling for messages, constant frequency transmission, transmission based on how much service usage or data traffic usage has taken place, transmission in response to device side control link message, service verification error events, other error events, and/or other message transmission trigger criteria can be determined, controlled and/or initiated by either the device side or the network side depending on the embodiment." *Id.*, 70:17-30.

606. Additionally, the '320 Patent makes clear that any component on the network (such as the server) and the (message-receiving) device can initiate the trigger. *See, e.g.*, *id.*, 70:21-30 ("For example, the timing functions, such as

asynchronous messages or polling for messages, constant frequency transmission, transmission based on how much service usage or data traffic usage has taken place, transmission in response to device side control link message, service verification error events, other error events, and/or other message transmission trigger criteria can be determined, controlled and/or initiated by either the device side or the network side depending on the embodiment.").

607. The '320 Patent discloses additional triggers for "the service control device link 1691," which are similar to the list of transmission/message delivery triggers described above. More specifically, the '320 specification states: "the service control device link 1691 determines when to transmit based potentially on several parameters including, for example, one or more of the following parameters: periodic timer trigger, waiting until a certain amount of service usage or traffic usage has occurred, responding to a service controller message, responding to a service controller request, initiated by one or more agents, initiated by a verification error condition, initiated by some other error or status condition." *Id.*, 38:30-38.

608. Other identified examples include a "transmission trigger [that] is based on waiting for an amount of service usage, and also including a minimum transmission rate that triggers a transmission according to one or more of the following parameters: a maximum time between transmissions clock to keep the service processor 115 in communication with the service controller 122 when little

or no service usage is occurring, a polling request of some kind from the service controller 122, a response to a service controller heartbeat, a transmission generated by a service verification error event, or a transmission generated by some other asynchronous event with time critical service processor 115 (or service controller 122) messaging needs, such as a transaction or service billing event or a user request." *Id*., 38:50-63.

609. The '320 specification provides additional details of the "service usage" or "service activity trigger," noting that it can be "based on some other measure than traffic usage, such as a number of messages transacted, one or more billing events, number of files downloaded, number of applications run or time that an application has been running, usage of one or more specified applications, GPS coordinate changes, roaming event, an event related to another network connection to the device and/or other service related measures." *Id*., 39:11-19.

610. For the above reasons, TS-23.140 discloses or renders obvious 1[d3].

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

611. TS-23.140 discloses or renders obvious that, upon determining that one of the message delivery triggers has occurred, the message buffer system includes logic for supplying one or more messages comprising the buffered content (e.g., stored message in the message store or MMSBox) to the transport services stack for delivery on the secure message link maintained between the transport services stack and a device link agent (e.g., MMS User Agent) on the given one of the wireless end-user devices.

612. As described in 1[d1]-1[d3] (which I incorporate here), TS-23.140 contemplates delivery of the stored message (in MMBox or another temporary storage) from the MMS Relay/Server upon the occurrence of one or more of the above-described delivery triggers. *See*, *e.g.*, Ex-1004, pp. 28-31, §7.1.3, 7.1.2.1. As I also described in 1[d3] (which I incorporate here), in my opinion, a POSITA would have found it obvious that MMS Relay/Server includes logic for performing message delivery upon detecting occurrence of one or more of the above-described delivery triggers.

613. Additionally, as described in 1[b1] (which I incorporate here), TS-23.140 discloses message delivery via the MMS Relay/Server's link interface (including the MM1 Transfer Protocol) that maintains secure message link (e.g., TCP/IP and TLS-based link) to a device link agent (e.g., a recipient MMS user agent) of a particular wireless end-user device (e.g., a UE/terminal corresponding to the recipient MMS User Agent). *See, e.g.*, *id.*, §6.2, p. 24, Fig. 4 (providing protocol framework to provide MMS; listing MM1 Transfer Protocol as part of MMS Relay/Server); Ex-1010, p. 22 (describing TLS-based link for data transmission between MMS User Agent and MMS Proxy-Rely in architectural configurations that employ HTTP based protocol stacks for MMSM implementation).

614. This limitation also references a new "transport services stack" for the first time that does not appear prior in the claim. In fact, comparing the familial '192 patent to the '320 patent, the '320 patent appears to use "link interface" in place of "transport services stack" for other claim limitations. *Compare, e.g.*, '192 Patent, cl. 1 ("***a transport services stack*** to maintain a respective secure message link") *with* '320 Patent, cl. 1 ("***a link interface*** to maintain a respective secure Internet data message link."). In my opinion, a POSITA would have understood that the claimed "transport services stack" is performing a similar function as the "link interface," such that the MM1 Transport Protocol and TCP/IP and TLS protocols—which

maintain a secure link between the MMS User Agent and the MMS Relay/Server—constitutes part of both a "link interface" and a "transport services stack".

615. For the above reasons, TS-23.140 discloses or renders obvious 1[d4].

i. ***1[e1]****: the device link agents on the respective wireless end-user devices, each of the device link agents configured to;*

616. TS-23.140 discloses a device link agent (e.g., MMS User Agent) on the respective wireless end-user devices.

617. As discussed above for 1[b1]-1[b2] (which I incorporate here), TS-23.140 discloses the network includes multiple UE/wireless devices that each host their own MMS User Agent to transmit and receive messages. Moreover, the MMS User Agents are configured consistent with 1[e2]-[e4], as I explain below.

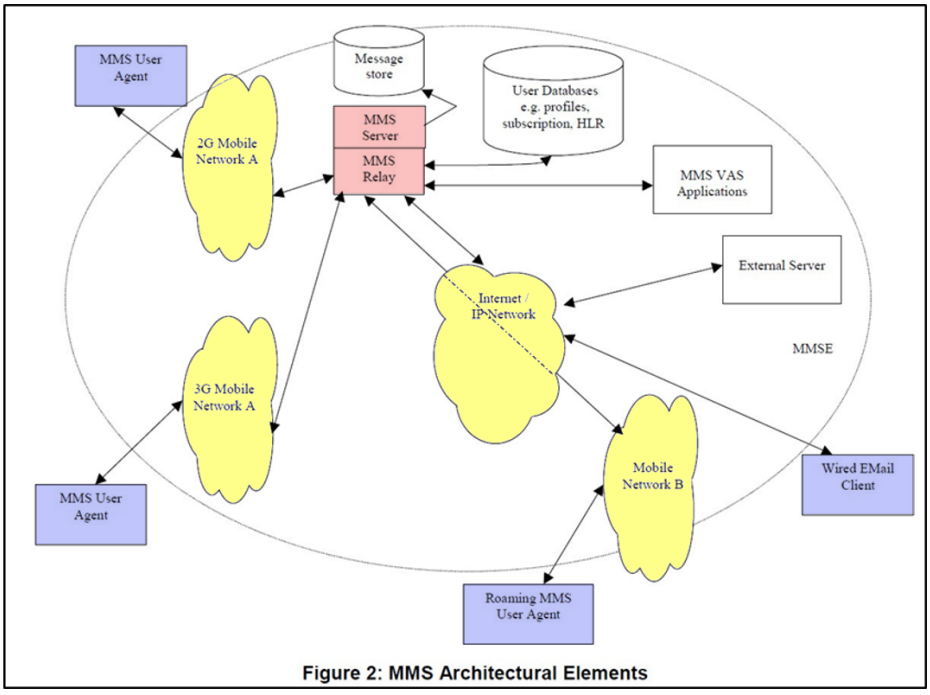


Figure 2: MMS Architectural Elements

618. For the above reasons, TS-23.140 discloses or renders obvious 1[e1].

j. ***1[e2]****: maintain the respective secure Internet data message link over a wireless network to the link interface;*

619. TS-23.140 discloses or renders obvious that the device link agent (e.g., MMS User Agent) is configured to maintain a secure internet data message link (e.g., secure link) to the link interface (comprising the MM1 Transport Protocol and TCP/IP and transport layer security (TLS)-based transport protocols).

620. As discussed above in 1[b1]-1[b2] (which I incorporate here), TS-23.140 discloses a TCP/IP link secured by TLS is established between the MMS Relay/Server and an MMS User Agent over an Internet network networking—including wirelessly—to transmit and receive MMs. A POSITA would have understood as a practical matter that, upon establishing a connection, it does not immediately terminate. Instead, it is maintained for some amount of time until some condition (e.g., a network failure, automatic timeout, or express connection release) occurs. *See, e.g.*, Ground 1D, below (discussing the use of heartbeat messages to prevent connection timeouts). For these reasons, a POSITA would have understood that the MMS User Agent "maintains" the secure connection for at least some period of time to successfully transmit and receive messages over the link.

621. For the above reasons, TS-23.140 discloses or renders obvious 1[e2].

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

622. TS-23.140 discloses or renders obvious that the device link agent (e.g., MMS User Agent) is configured to receive secure internet data messages (e.g., MMS messages) from the network server system (e.g., MMS Relay/Server) over a secure internet data message link (e.g., link secured by TLS), including messages collected from multiple network elements (e.g., MMS User Agents resident on MMS client devices, MMS VAS Applications) and messages corresponding to multiple software components (e.g., applications) authorized to receive messages via the device link agent (e.g., MMS User Agent) on that respective device, where a subset of messages (e.g., application MMs) include both a unique identifier (e.g., application identifier) for a corresponding software agent (e.g., application), and data to be consumed (e.g., application data) by that software component provided by a respective network element (e.g., MMS User Agents resident on MMS client devices, MMS VAS Applications) corresponding to that software component (e.g., application).

623. In detail, as I explained previously in 1[b1]-[d4] (which I incorporate here), TS-23.140 discloses the MMS User Agent receives secure messages from the MMS Relay/Server utilizing the MM1 Transport Protocol with TCP/IP and TLS protocols to maintain a secure link between the two. Because messages may be transmitted over TLS—a "secure Internet data message link," as I address in 1[b1], also incorporated here)—in my opinion, a POSITA would have understood messages sent over that secure link would constitute "secure Internet data messages" as claimed. Moreover, as I also explain in 1[b1]-[c2], TS-23.140 also discloses MMS User Agents may receive messages from a variety of other User Agents, VAS Applications, etc. (e.g., multiple network elements), and those messages correspond to multiple applications registered with the User Agent on each device. The relationships between these various network elements are exemplified in the figure below.

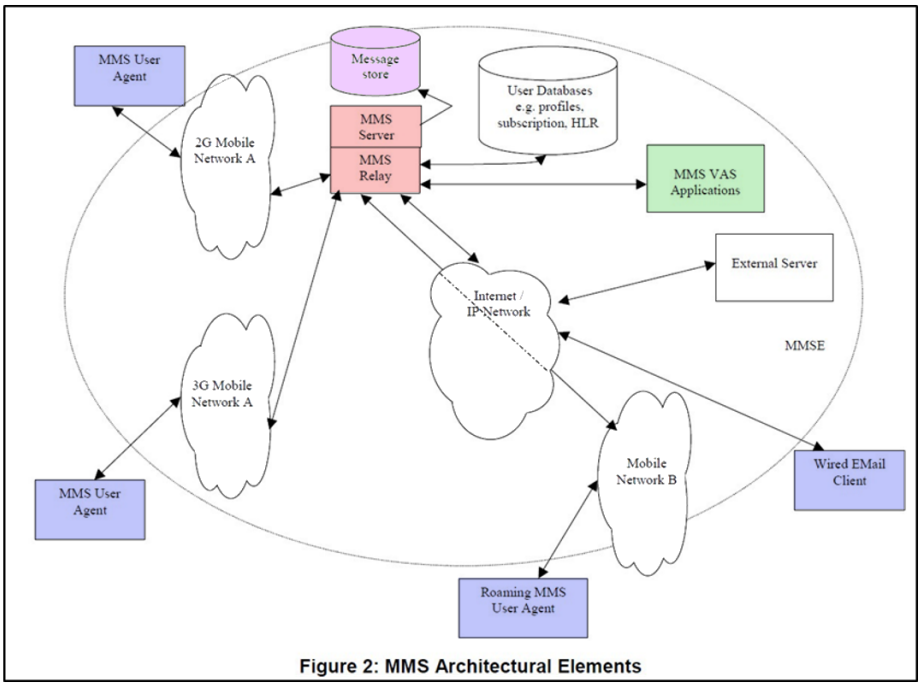


Figure 2: MMS Architectural Elements

Ex-1004, § 4.2, Fig. 2, p. 17 (describing the above MMS environment as "***a collection of MMS-specific network elements***.").

624. TS-23.140 also explains that when a registered application transmits an application-specific MM, the MM includes application data, "control information", and a destination "application identifier". *Id.*, 54 ("MMS may also be used to transport data specific to applications. Applications that intend to transport application specific data using MMS may either reside on an MMS User Agent or on an MMS VAS Application."), 55 ("The application identifier of the destination application shall be present in an abstract message…. [T]he MMS User Agent or MMS VAS Application may receive information from the application…[and] may insert this information in both the information elements and/or payload (if present)

of the abstract message."), 55-56 (describing actions upon an MMS User Agent receiving an application message), 14.

625. Taken together, a POSITA would have understood that application-specific MMs sent over the secure link are "a subset of secure Internet data messages" as claimed, which include both a "unique" destination application identifier to identify the intended application and application data intended for that application supplied by the originating network element.

626. For the above reasons, TS-23.140 discloses or renders obvious 1[e3].

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

627. TS-23.140 discloses or renders obvious that, for software components (e.g., applications) authorized to access messages received by the device link agent (e.g., MMS User Agent), causing messages with a unique identifier corresponding to a software application (e.g., application messages with application identifiers) to be securely delivered to a software process (e.g., the application processes) corresponding to the given software component (e.g., application).

628. In detail, as I explained in 1[b1]-1[c2] and 1[e3] (which I incorporate here), TS-23.140 discloses the MMS Relay/Server transmits application-specific MMs—which contain application identifiers to registered applications—to MMS

User Agents using MM1 Transport Protocols and TCP/IP and TLS protocols for a secure link, and then the MMS User Agent routes the message to the corresponding application on the UE/mobile device. TS-23.140 also discloses that, to transmit over MMS (and as discussed above in 1[c1]), the application must register with the MMS User Agent, which then sends the transmitted message to its destination. Ex-1004, 54-56. The destination application is also a registered application that has undergone the same registration process with the requisite MMS User Agent. *Id*.

629. In more detail, TS-23.140 discloses that applications, such as those that are downloaded on a mobile phone, can use MMS to "transport data specific to ***applications***" and MMS communication can include "transport data specific to applications between two MMS User Agents or an MMS User Agent and an MMS VAS Application." *Id.*, §7.1.18, pp. 54-55. Thus, TS-23.140 discloses that each UE (wireless end-user device) includes multiple software applications (software components). Ex-1036, 2-4 (showing and listing multiple applications in UEs that receive application data via MMS), Fig. 5 below.

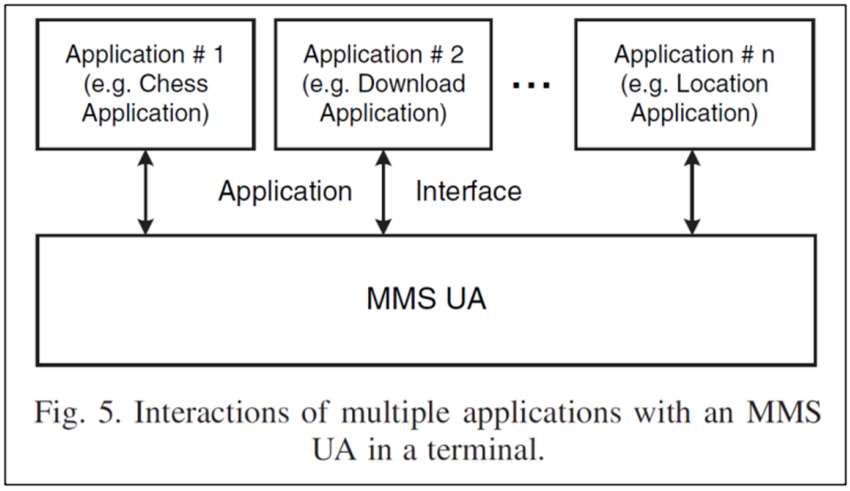


Fig. 5. Interactions of multiple applications with an MMS UA in a terminal.

630. A POSITA would have understood or found it obvious that each such application has to be authorized to send messages via MMS. TS-23.140 discloses that "[a]pplications that want to transport data specific to applications other than MMS will initially ***need to register with the appropriate MMS User Agent or MMS VAS Application***." Ex-1004, §7.1.18, pp. 54-55. "During this registration process the application provisions an MMS User Agent or an MMS VAS Application with its application identification value and may negotiate with the MMS User Agent or MMS VAS Application the details (amount and format) of information to be exchanged between the two entities." *Id.*

631. Additionally, the MMS User Agent can indicate the "Terminal Capability" toward its recipient MMS Relay/Server by transmitting "a set of information describing the terminal's capabilities, including a binary flag indicating whether or not the terminal supports Application data." *Id.*, §7.1.3.1, p. 30.

632. Subsequently, a transmitted message including application data for an intended application is only delivered to that application upon determining that the MMS User Agent has the capability of supporting application data and the terminal includes the intended application for the receipt of such application data. *Id.*, pp. 30, 54-56 ("If the MMS Relay/Server finds from the recipient MMS User Agent's capability indication … that the recipient MMS User Agent supports transport of application data, the MMS Relay/Server … shall pass on the destination application identifier, the 'reply-path' identifier (if present) and the additional application/implementation specific control information (if present) unaltered."; "If the destination application does not reside on the receiving MMS User Agent or on the USIM or MMS VAS Application, the MMS User Agent or MMS VAS Application shall discard the corresponding abstract message."). From these disclosures a POSITA would have understood that such applications are ***authorized*** to send messages including application data to specific applications on other devices.

633. In my opinion, therefore, a POSITA would have understood that the application-specific MMs are specifically directed to their corresponding applications (e.g., a VAS Application running on a server may send a message tailored to its corresponding application). A POSITA would have understood that the contents of these messages would be directed to a "software process," as claimed, to utilize the application data transmitted in the MM. That is the entire point of

application-specific MMs; without being processed, there would be no use of enabling application-based functionality in the MMS network.

634. For the above reasons, TS-23.140 discloses or renders obvious 1[c2].

> ***2. Claim 2:*** *The networked system of claim 1, the message buffer system logic to determine, for each of the wireless end-user devices, when one of a plurality of message delivery triggers has occurred, at least one of the triggers for each given device specific to one or more states of that given device.*

635. TS-23.140 discloses that one of the message delivery triggers is the state of the given device.

636. As discussed above in 1[d3]-1[d4], TS-23.140 discloses or renders obvious that message buffer system logic determines when a message delivery trigger has occurred. As I also explained in those sections, when the MMS User Agent is unavailable to receive a message, an indication (trigger) that the MMS User Agent has later become available/reachable (e.g., moves into coverage, switches User Agent on) serves to initiate a delivery attempt. *Id.*, 29 (§7.1.3). For instance, if a user is out-of-network coverage, or disables the User Agent on the wireless device, the User Agent would not be "reachable" by the MMS Relay/Server. When the User Agent becomes reachable, however, that triggers message delivery.

637. The '320 Patent specification itself acknowledges that the operational status of the wireless device (e.g., whether the device is online or offline) constitutes a "state" of the device. Ex-1001 ('320 Patent), 108:19-20; 180:40.

638. For the above reasons, TS-23.140 discloses or renders obvious claim 2.

3. Claim 3: *The networked system of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

639. TS 123.140 discloses or renders obvious that one of the message delivery triggers is the expiration of a periodic timer.

640. In more detail, TS-23.140 discloses that the originating MMS User Agent can set the desired time of delivery of the MM, in which case the MM is stored by the MMS Relay/Server until the ***earliest desired time of delivery***, before the message is delivered. *See* Ex-1004, § 7.1.1, p. 27.

641. In my opinion, a POSITA would have understood or found it obvious that specifying the desired time of delivery triggers a periodic timer upon the receipt of a message that expires when the delivery time is reached, which then triggers the delivery of the message. Such a timer would be periodic in that each time a message is received, the timer is triggered and continues for a particular period at the expiration of which, the message is delivered to the intended recipient.

642. Additionally, or alternatively, MMS User Agent uses periodic polling to identify new messages on an external server, and subsequently retrieves those messages—which also demonstrates another example a message delivery trigger that is the expiration of a periodic timer (as indicated by periodic polling). *See id.*, §8.3.3, pp. 90-91 (explaining that discovery of new messages on external servers can be done by "***periodic polling*** for messages on External Server, followed by retrieval

by the MMS User Agent via the MMS/Relay Server"); §3.1, p. 14 (explaining that an external server is a "network entity/application of an external system such as Internet email, unified messaging system or facsimile to which MMs may be sent to and/or from which MMs may be received by an MMS User Agent via an MMS service provider").

643. A POSITA would have understood that (1) the above-described periodic polling for messages is akin to the claimed feature of a periodic timer expiring because the periodic polling would be done at periodic intervals (i.e., after a certain time period has elapsed/expired) that repeat regularly and (2) this periodic polling is a message delivery trigger because, in response to receipt of such polling, messages from the respective servers would be provided/delivered to the MMS User Agent via the MMS Relay/Server. When the Relay/Server receives the message at polling timer expiration and the earliest desired time of delivery is not specified—the polling timer's expiration starts a process that attempts message delivery.

644. For the above reasons, TS-23.140 discloses or renders obvious claim 3.

4. ***Claim 5:*** *The networked system of claim 1, wherein one of the message delivery triggers is the receipt of a transmission on the respective secure Internet data message link from the device link agent of the given one of the wireless end-user devices, or a response generated to a transmission received from that device link agent.*

645. TS-23.140 discloses that one of the message delivery triggers is the receipt of a transmission over a secure link from the device link agent.

646. In detail, as described in 1[b1], 1[d1]-[d2], and 1[d3] (which I incorporate here), TS-23.140 discloses that one of the message delivery triggers is a manual request for delivery from the MMS Relay/Server over a TLS-secured link in response to a notification from the MMS Relay/Server. Ex-1004, 28-29. In other words, the MMS Relay/Server *receives* a transmission (a request for manual delivery) from the MMS User Agent over the secure link between the two to trigger a message delivery.

647. For the above reasons, TS-23.140 discloses or renders obvious claim 5.

5. ***Claim 6:*** *The networked system of claim 1, wherein one of the message delivery triggers is a heartbeat message generated by the given device link agent, or a request received from the given device link agent.*

648. TS-23.140 discloses one of the message delivery triggers is a request received from a device link agent.

649. In detail, as I explained previously in 1[d3] (which I incorporate here), TS-23.140 discloses that one of the message delivery triggers is a manual request for delivery from the MMS Relay/Server over a TLS-secured link (secure message link). *See also id.*, §7.1.2.1, p. 28-29.

650. For the above reasons, TS-23.140 discloses or renders obvious claim 6.

6. ***Claim 7:*** *The networked system of claim 1, wherein one of the message delivery triggers is the receipt of a particular message from one of the network elements.*

651. TS-23.140 discloses one of the message delivery triggers is the receipt of a particular type of message from a network element.

652. In detail, as I explained previously in 1[d3] (which I incorporate here), TS-23.140 discloses that the message delivery triggers may include (among others) receipt of a message that does not specify an earliest desired time of delivery, receipt of a manual/automatic retrieval request, or receipt of a message indicating that the MMS User Agent has become available/reachable.

653. In this manner, a POSITA would have understood that TS-23.140 contemplates message delivery triggers that include receipt of certain types of messages from one or more network elements (originating MMS User Agents operating of UEs/client devices).

654. For the above reasons, TS-23.140 discloses or renders obvious claim 7.

> ***7.*** ***Claim 10:*** *The networked system of claim 1, wherein a second subset of the secure Internet data messages contain a user message from a network element, the user message intended for display on a user interface of a given one of the wireless end-user devices.*

655. TS-23.140 discloses a subset of the secure Internet data messages (e.g., MMS messages) contain a user message from a network element intended to be displayed on a user interface of the user's wireless device.

656. First, as I explained previously in 1[e3] (and incorporate here), TS-23.140 discloses sending "secure Internet data messages" as claimed between network elements, including the MMS Relay/Server and MMS User Agent.

657. TS-23.140 also discloses that at least some of those messages "may be displayed to the end user with or without any pre-notice," and notes that MMs shall support specified types of media, including plain text, speech, audio, images, bitmap graphics, video, and others. TS-23.140, 20. In my opinion, a POSITA would have understood that many of these types of media are specifically intended to be displayed to a user. For instance, the entire purpose of providing MMS was to allow users to send multimedia (such as images and video) over a cellular network, rather than just plain text like existing SMS systems at the time. For these reasons, in my opinion, a POSITA would have understood that messages of these types of file formats intended to be displayed to a user would be considered "user messages" as claimed, as opposed to some other type of back-end message (like configuration/parameter messaging) that may be used by other software elements. This is in contrast, however, to application MMs intended for application processes, which are delivered to an application "without presentation to the user." *Id.*, 56.

658. For the above reasons, TS-23.140 discloses or renders obvious claim 10.

8. ***Claim 11:*** *The networked system of claim 1, wherein one of the software components on a given one of the wireless end-user*

devices is a policy control agent, and the data to be consumed by the policy control agent comprises service settings and/or configuration information for the given device.

659. TS-23.140 discloses one software component on a wireless device is a policy control agent that consumes service setting or configuration information for that device.

660. In detail, TS-23.140 discloses the MMS network includes an MMS User Databases with "elements that…contain user related information such as subscription and configuration" data, user profiles, access controls, server storage space, delivery rules, and UE/wireless device capabilities. *Id.*, 18, 20. TS-23.140 also shows that MMS User Agents may interface with the MMS User Databases (through the MMS Relay/Server) using at least MM1 and MM6 Transport Protocols.

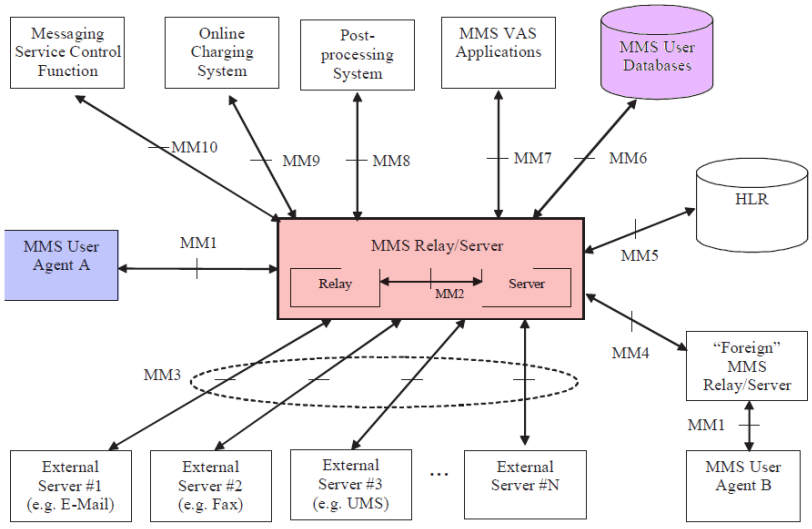


Figure 3: MMS Reference Architecture

Id., Fig. 3.

661. In my opinion, a POSITA would have understood that settings and policies are routinely updated and communicated between a user's wireless device and a back-end system. *See, e.g.*, Ex-1041, [0032] ("The MM preferably ***manages a user database and contains the update status of each one of the endpoints 156/166*** as well as their connection parameters and capabilities."); Ex-1046, [0032] ("The one or more databases 134 may comprise one or more entities that contain user related information such as subscription and configuration (e.g. user profile, subscription, operator services, Home Location Register ('HLR'), etc.) and provide customized processing instructions. The one or more databases 134 may provide MMS user subscription information, information for the control of access to the MMS, information for the control of the extent of available service capability (e.g. server storage space), a set of rules how to handle incoming messages and their delivery, and information of the current capabilities of the user's terminal."); Ex-1042, [0043] ("the wireless service provider [may] update the relevant database so that the user's MSISDN number points to the address of the MMS server belonging to the ISP."). For instance, it was well-known for service or application providers to push (through messages) updated settings/configuration information to a user's wireless device. *See, e.g.*, Ex-1048, [0003] ("It is known to configure the mobile device with the WAP gateway address and WAP port number via the user interface of the mobile device.… In OTA programming, ***the service provider can transmit an***

SMS configuration message to the mobile device whereupon the mobile device configures itself with the WAP gateway address and port number included in the SMS configuration message."). Moreover, MMS permitted users to establish their own delivery rules, it was also well-known for users to upload their own rules to the User Database. *See, e.g.*, Ex-1047, [0037] ("the respective recipient rules can be uploaded or otherwise transferred to the respective databases for storage. For example, the recipients can transmit the recipient rules to the profile server, which thereafter stores the recipient rules in the database associated with the profile server.").

662. A POSITA would have therefore understood that the device hosting the MMS User Agent would have some hardware or software element(s) (e.g., a "policy control agent" as claimed) to process these various setting and configuration elements capable of being transmitted over MMS, including service settings and/or configuration information for a device. That way, any changes to a user profile, service settings, access controls, delivery rules, and UE/wireless device capabilities can be updated on both, e.g., the MMS User Database (on the back end) and the user's UE/wireless device (on the front end) to implement any settings changes.

663. For the above reasons, TS-23.140 discloses or renders obvious claim 11.

9. Claim 15

*a. 15[a]-15[b]: [15a] wherein the device link agent on a given one of the devices is further configured to receive, from one or more of the software components on that device, upload messages, **[15b]** each of the upload messages identifying a corresponding one of the network elements to which the device respective software component has requested delivery*

664. TS-23.140 discloses or renders obvious that the device link agent (e.g., MMS User Agent) is configured to receive upload messages (e.g., application messages) from software components (e.g., applications) on that device, with each upload message identifying the corresponding network element (e.g., MMS User Agents resident on other MMS client devices, MMS VAS Applications) to which the software component (e.g., application) has requested delivery.

665. In more detail, TS-23.140 discloses that an application executing on a user device/UE (i.e., a device software component) can forward a message including application data (i.e., upload message) to the UE's MMS User Agent (i.e., the device link agent), which in turn can submit the message to the MMS Relay/Server. Ex-1004, §7.1.18, pp. 54-55; § 3.1, p. 14. TS-23.140 also discloses that the MMS User Agent can submit/forward to the MMS Relay/Server MMs with requests to store them. *Id.*, 40.

666. A POSITA would have understood that an application receiving application data using MMS can also send application data to an application resident

on another device (e.g., another device with an MMS User Agent or an MMS VAS Application).

667. For example, TS-23.140 discloses that "an application may trigger an MMS User Agent or an MMS VAS Application to submit certain abstract messages. Upon triggering an MMS User Agent or an MMS VAS Application to send an abstract message the MMS User Agent or MMS VAS Application may receive information from the application. The MMS User Agent or MMS VAS Application may insert this information in both the information elements and/or payload (if present) of the abstract message." *Id.*, 55.

668. Both the MMS User Agent and the MMS Relay/Server facilitate "transport of application data." *Id.*, §§5.1.1 and 5.1.2, pp. 19-21, 35 (describing transport of application data by MMS User Agent to another MMS User Agent via the MMS Relay/Server), 55 (transporting data between applications of two MMS User Agents). Moreover, as described with reference to 1[b1] (which I incorporate here), the MMS Relay/Server includes the MM1 Transfer Protocol that facilitate receipt/transport of application data from the MMS capable UE/MS to the MMS Relay/Server. *Id.*, §6.2, p. 24, Fig. 4. From these disclosures, a POSITA would have understood or found it obvious that the transport stack of the MMS Relay/Server receives message including application data from an application via a particular MMS User Agent.

669. Moreover, in the MMS Environment contemplated by TS-23.140 (*see* Figs. 1 and 2), a POSITA would have understood that multiple different devices, including end-user devices/terminal, would be able to use an MMS Relay/Server for transmission of data to multiple other devices/server/network elements.

670. In such an environment, a POSITA would have found it obvious for the MMS Relay/Server to receive upload messages (including application data) from multiple MMS User Agents and their respective applications requesting delivery of such data, and such communications would happen over the dedicated TLS-based links between each such MMS User Agent and the MMS Relay/Server (as described with reference to 1[b1]-1[c2].

671. TS-23.140 further renders obvious each of the upload messages identifying a corresponding one of the network elements (e.g., recipient MMS User Agents on UEs) to which the device respective software component (e.g., originating application on a sending UE) has requested delivery.

672. As described above in 1[c1] (which I incorporate here), the application data is included in the upload message (i.e., the abstract message) from an originating MMS Agent and its associated device, and this abstract message further includes the "application identifier of the destination application." *Id.*, §7.1.18, pp. 54-56 ("Abstract messages that are sent by an MMS User Agent or an MMS VAS Application on behalf of an originating application shall contain a destination

application identifier."). Because the message includes an identifier of a destination application resident on a particular device/terminal, a POSITA would have understood that upload message identifies the corresponding device/terminal (i.e., the network element).

673. Additionally/alternatively, as with sending any multimedia messages (MMS), including MMs that include application data, the sender MMS User Agent "indicate[s] the address of the MM recipient." *See id.*, § 7.1.1, p. 26; *see also* pp. 90, 190 (showing data types included in messages sent over MM1 interface between MMS User Agent and the MMS Relay/Server, including "Recipient address").

674. Indeed, because messages are transmitted from a MMS User Agent to a recipient MMS User Agent (or MMS VAS Application) (on another device/server), in my opinion, a POSITA would have found it obvious for the message to include an identifier or address for the device/server to which the message is being transmitted.

675. A POSITA would have understood or found it obvious that routing messages to particular devices would be facilitated when including some device identification to facilitate that routing over a network. Indeed, TS-23.140 discloses that for MM1 communications (between MMS Relay/Server and MMS User Agents), the message must be addressed using three addresses: "the address of the MMS Relay/Server, the address of the recipient and the address of the originator."

Id., § 7.2.1, p. 57. "The recipient's address can be a user's address, a user's terminal address, or a short code." *Id.* Additional details about message addressing and formats are provided below (*id.*):

676. The MMS User Agent and MMS Relay/Server shall support both E.164 (MSISDN) and RFC2822 addressing formats. The reference point MM1 should support a way to indicate the used address type to enable future extension. The encoding of the addressing is up to the corresponding MM1 implementation (cf. Annex B).

E.g. the originator MMS User Agent may specify each of the address fields in one of the following formats:

4) RFC 2822 address (FQDN or unqualified)

5) PLMN address: ["+" | "" | "#"] [digit | "*" | "#"] ... ["/TYPE= PLMN"]*

6) Other "/TYPE= "

The "/TYPE= " field specifies the address type. When PLMN format is used the type is optional. The "/TYPE= " convention provides flexibility for future enhancements.

When the "/TYPE=" qualifier is absent, the MMS Relay/Server should resolve potential ambiguities by applying the following logic to the address in the following order:

4. if it contains the "@" character, the address should be interpreted as an FQDN RFC2822 address

5. if it is completely numeric, except possibly including "+", "", or "#", it should be interpreted as "/TYPE= PLMN", e.g. an E.164 address, a local telephone number, or a numeric short code,*

6. otherwise, it should be interpreted as an unqualified RFC2822 address (alphanumeric short code)

677. Thus, a POSITA would have found it obvious to include such addressing in the upload message to facilitate transmission of data between devices of a network. Thus, given TS-23.140's disclosures and well-known device addressing aspects for network communications, a POSITA would have found it obvious to include such addressing in the upload message for transmitting data between network devices.

678. For the above reasons, TS-23.140 discloses or renders obvious 15[a]-15[b].

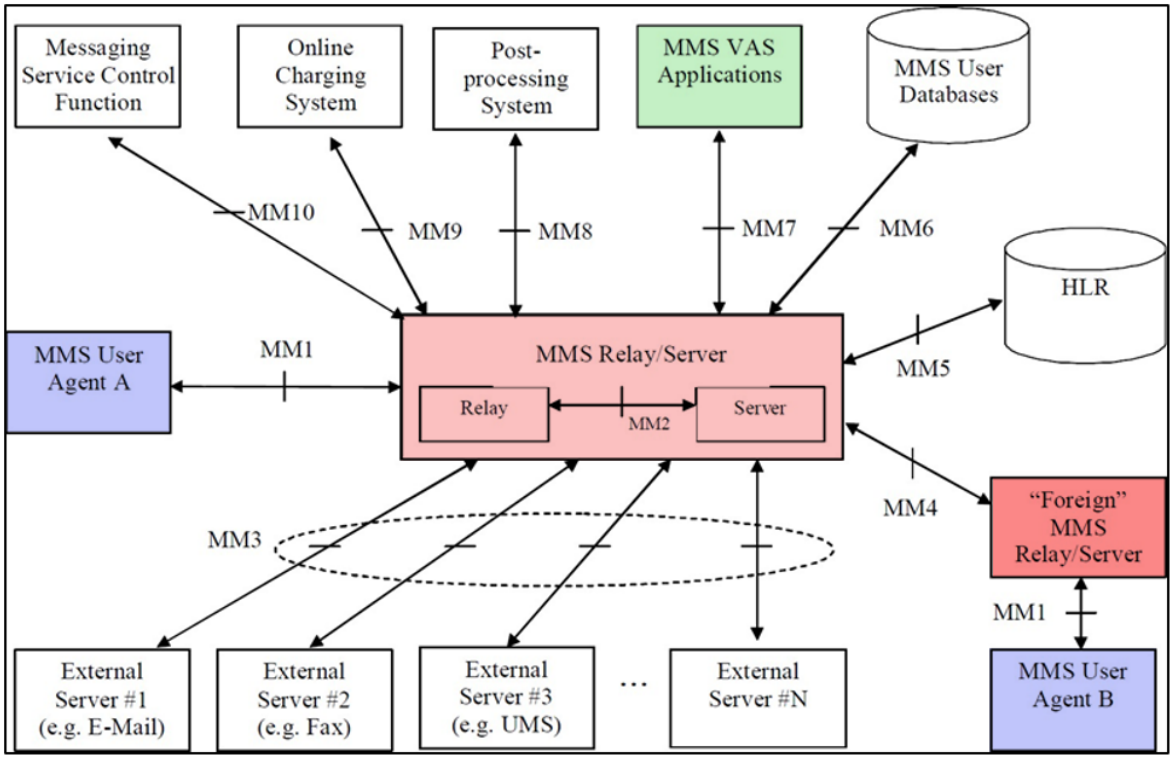
b. ***15[c]:*** *the device link agent on that device transmitting the upload messages to the network message server over the respective secure Internet data message link, for delivery by the network message server to the respective identified network elements*

679. First, I note that claim 15 uses the term "network message server" that does not appear elsewhere in the claims. However, I have assumed that the "network message server" refers to the "network server system" that appears in claim 1. With that understanding, TS-23.140 discloses or renders obvious that the device link agent

(e.g., MMS User Agent) on a device transmits the upload messages to the network message server (e.g., MMS Relay/Server) over the secure Internet data message link (e.g., link secured by TLS) for delivery by the network message server to the identified network element(s).

680. As I described in 1[b1] and 15[a]-[b] (and incorporate here), TS-23.140 discloses the MMS User Agent transmits application-specific messages to the MMS Relay/Server across a (secure) link using the MM1 Transport Protocol and TCP/IP and TLS Protocols. And as I explained in 1[c1] and 1[c2] (and incorporated here), the MMS Relay/Server then delivers the message to the identified recipient.

681. In my opinion, a POSITA would have understood or found it obvious that the MMS Relay/Server transports application data between an MMS User Agent and another MMS User Agent/VAS Application. Ex-1004, 19-21, 35, 55-56 (describing the process of sending/receiving application MMs to/from an MMS User Agent through the MMS Relay/Server). The MMS Relay/Server implements, for instance, the MM1 Transfer Protocol to receive application data received from an application on a UE/mobile device and transport it to another MMS-capable UE/mobile device; the MM7 Transport Protocol to transport it to an MMS VAS Application; or the MM4 Transport Protocol to transport it to another UE on a foreign Relay/Server. *See, e.g., id.*, 23-24, Figs. 3-4.



682. For the above reasons, TS-23.140 discloses or renders obvious 15[c].

10. ***Claim 18:*** *The networked system of claim 1, wherein at least one of the secure Internet data messages comprises multiple identifier/data pairs.*

683. As described below, TS-23.140 discloses or renders obvious that at least one of the one or more secure Internet data messages (e.g., MMS messages) comprises multiple identifier/data pairs.

684. As I described in 1[b1], 1[c1]-1[c2], and 1[d1]-1[d4] (which I incorporate here), the MM1 link interface that uses TCP/IP an TLS is used to communicate between an MMS User Agent (or VAS Application) and an MMS Relay/Server, and an originating MMS User Agent (or an originating MMS VAS Application) sends, via the MMS Relay/Server, application data as well as the

identifier of the destination application to the destination application corresponding to a destination MMS User Agent. Thus, such a message includes an identifier/data pair in the form of the application data and the destination application identifier. Ex-1004, 54-56.

685. As I also explained in 1[c1]-1[c2], in my opinion, a POSITA would have understood that multiple registered applications on a UE/mobile device would receive messages including application data via the MMS Relay/Server. *See, e.g.*, *id.*, 54-56; Ex-1036, 3-4.

686. In my opinion, a POSITA would have understood that, in some situations, application data/identifiers for multiple applications should be consolidated in a single message. For instance, TS-23.140 discloses that one delivery trigger is when the MMS User Agent becomes available/reachable (Ex-1004, §§7.1.2.1-7.1.3, pp. 28-30), meaning multiple messages may be queued for delivery at one time. In these circumstances, a POSITA would have been motivated to consolidate application data and corresponding application identifiers for multiple applications in a single message to achieve network efficiencies compared to transmitting multiple separate messages. *See, e.g.*, Ex-1039, Abstract (reducing number of messages signaled over a network "to reduce CPU utilization at both the source and the target system, which improves the overall system throughput and performance, and also reduces the bottleneck at the network level."). A POSITA

would have had a reasonable expectation of success in making this modification because it merely requires combining existing data structures.

687. Additionally, the concepts of broadcast (https://web.archive.org/web/20201128164630/https://csrc.nist.gov/glossary/term/broadcast) and multicast (Tom Bachert, IPv4 Multicast Security: A Network Perspective, SANS Institute, https://sansorg.egnyte.com/dl/rtTCC6TbTpFq) messaging are applicable, wherein a single address can be used to send messages to many or all devices. As such, a POSITA would have understood or found it obvious that there were multiple known techniques for delivering messages to multiple recipients (thereby providing messages with multiple identifier/data pairs that are intended for applications—where data for application is routed to that application using its identifier included in the message(s)).

688. For the above reasons, TS-23.140 discloses or renders obvious claim 18.

B. Ground 1B: Claim 3 Is Rendered Obvious by TS-23.140 and Adamczyk

1. ***Claim 3:*** *The networked system of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

689. TS-23.140-Adamczyk renders obvious a message delivery trigger being the expiration of a periodic timer.

690. Adamczyk discloses a message-transmission system like TS-23.140. *See, e.g.*, Ex-1021, Abstract, [0007]-[0009]. Adamczyk, however, expressly describes how notification servers (like TS-23.140's MMS 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 the preferences of the user. *Id.*, [0010], [0022], cl. 1.

691. Although TS-23.140 discloses a series of automatic and manual triggers (which I describe in 1[d3] and incorporate here), in my opinion, a POSITA would have appreciated that TS-23.140's (automatic) triggers could be supplemented with the additional types of triggers like a periodic schedule expressly disclosed by Adamczyk.

692. In my opinion, a POSITA would have been motivated to add an additional periodic timer trigger to the list of triggers that TS-23.140 already discloses for several reasons. First, additional message delivery triggers create a more configurable system that better suits a wider range of user needs in an efficient manner. For instance, rather than a recipient MMS User Agent specify a single deferred time of delivery, or require the MMS User Agent to perform periodic polling to retrieve messages from external networks, a POSITA would have been motivated to implement a periodic timer like Adamczyk's to have the MMS

Relay/Server issue notifications at pre-determined intervals will all messages received during that interval.

693. Second, a POSITA would have appreciated that periodic timers would arguably improve the battery life of the device operating the MMS User Agent. Rather than have to establish a connection between the MMS User Agent and MMS Relay/Server every time a message is received by the MMS Relay/Server (causing a notification to be sent to the MMS User Agent when in automatic retrieval mode and no deferred delivery), or have the MMS User Agent repeatedly ping the MMS Relay/Server to see if messages are pending, the number and timing of connections and retrievals could be limited to pre-set intervals. *See, e.g.*, Ex-1040, [0074] (describing battery benefits of setting intervals for call connections rather than persistent connections).

694. In my opinion, a POSITA would have had a reasonable expectation of success in adding the periodic timer trigger of Adamczyk to the TS-23.140 system because the references suggest such a combination. As discussed above, Adamczyk expressly recognizes that its teachings can be used to augment existing 3GPP notification networks. Ex-1021, [0017]. Additionally, TS-23.140 already discloses a periodic polling method of retrieval in which a UE regularly requests retrieval from an MMS Relay/Server. Augmenting the Relay/Server to include periodic notification pushes based on some configurable timer would be another

implementation that would include the above noted benefits. Configuring the Relay/Server to push messages at predetermined, periodic intervals, would have merely required configuring the Relay/Server to hold received messages in the memory store/MMBox until expiry of a regularly recurring timer or periodic scheduled event before pushing all messages received during the time interval at once.

C. Ground 1C: Claim 4 is Rendered Obvious by TS-23.140 and Herzog, With or Without Adamczyk

1. ***Claim 4:*** *The networked system of claim 3, wherein the period of the timer is fractionally shorter than a maximum data message interval beyond which the Internet data message link to the given device is taken down.*

695. TS-23.140-Herzog, or TS-23.140-Adamczyk-Herzog renders obvious setting the period of the timer fractionally shorter than a maximum data-message interval after which the secure connection is disabled.

696. As I described previously in two assessments of claim 3 (one in view of TS-23.140 alone, and another in view of TS-23.140 and Adamczyk, incorporated here), TS-23.140 describes a periodic timer through periodic polling, and TS-23.140-Adamczyk together teach a periodic schedule for message delivery. These references do not, however, appear to expressly describe what happens to the connections between the MMS User Agent and MMS Relay/Server to account for these periodic schedules.

697. Herzog is one example of prior art, however, that taught mechanisms to control the duration of a connection between a client (like TS-23.140's MMS User Agent) and server (like TS-23.140's MMS Relay/Server) through keep-alive timers to periodically ping a server to keep the connection alive, or to terminate the connection when the keep-alive messages are not sent for a given period of time. Ex-1008, [0036], [0041]-[0043].

698. In my opinion, a POSITA would have been motivated to implement keep-alive timers and messages in the TS-23.140 and/or TS-23.140-Adamcyzk system to control the duration and termination of a connection between the MMS User Agents and MMS Relay/Server automatically. Specifically, connections would be maintained for as long as the MMS User Agent indicates it is available and would automatically disconnect when the MMS User Agent stops sending keep-alive messages. Incorporating a keep-alive timer would have several benefits. First, the MMS Relay/Server would be able to track the availability and connection status of each MMS User Agent. Second, the MMS Relay/Server would avoid trying to maintain connections with disconnected/unavailable MMS User Agents. This would, in turn, reduce bandwidth demand and provide openings for other MMS User Agents to connect to the MMS Relay/Server to the extent the availability of a connection was a bottleneck (e.g., during periods of high demand).

699. In my opinion, a POSITA would have had a reasonable expectation of success in making this combination because TS-23.140 discloses a similar messaging system to Herzog, including a gateway intermediary between a server and a client. Ex-1008, [0005], [0024] (explaining that the server "is any type of server that is programmed to push content to the client"); Ex-1004, 30. And like TS-23.140, Herzog discloses that messages can be "saved in a message buffer…until [the message] can be delivered to the client" (e.g., when the connection is kept open by keep-alive messages). Ex-1008, [0031]. Herzog also discloses using various types of connections, including IP, TCP, SSL and HTTP. *Id.*, [0026], [00037]. As a result, the combination would only require implementing Herzog's keep-alive-messaging between the MMS User Agent and MMS Relay/Server to maintain TS-23.140's existing connection.

700. Moreover, upon making this combination, in my opinion, a POSITA would have understood that the period of the periodic polling (disclosed by TS-23.140) or periodic message delivery schedules (taught by TS-23.140 combined with Adamczyk) would be some degree shorter than the period required for the keep-alive timers to indicate availability of the MMS User Agent. If the periodic polling/message delivery were set to a longer period than the keep-alive timer period, the connection may be terminated before the periodic polling/delivery message could be sent, thus requiring a new connection to be established. And

because the ’320 Patent does not specify what constitutes “fractionally shorter,” in my opinion, any degree shorter can be considered “fractionally shorter,” as claimed, since it would be some “fraction’ of the duration of the keep-alive timers.

D. Ground 1D: Claims 6, 12, 13, and 16 are Rendered Obvious by TS-23.140 and Shen

1. ***Claim 6:*** *The networked system of claim 1, wherein one of the message delivery triggers is a heartbeat message generated by the given device link agent, or a request received from the given device link agent.*

701. TS-23.140-Shen renders obvious that the transmission is a heartbeat message generated by the given device link agent, or a request received from the given device link agent.

702. TS-23.140 discloses one trigger is when a recipient MMS User Agent becomes available/reachable. Ex-1004, 29.

703. Shen further teaches that a heartbeat message generated by a user’s device is used to indicate reachability of the device to receive stored messages by pinging the gateway at intervals to express its reachability/availability. Ex-1014, [0033]-[0034].

704. In my opinion, a POSITA would have been motivated to incorporate Shen’s “heartbeat” evaluation as an additional trigger in TS-23.140 by providing a specific mechanism (a heartbeat message) to trigger message delivery in a way that TS-23.140 already envisions (based on reachability of the MMS User Agent). For

this reason, in my opinion, a POSITA would have been motivated to implement a heartbeat message generated by the MMS User Agent to ping the MMS Relay/Server in TS-23.140 at intervals because this would allow the MMS Relay/Server to periodically send buffered messages to the UE/mobile device when the device is available/reachable.

705. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification not only because Shen envisioned operating in an MMS network like TS-23.140 (which I described in claim 2 and incorporate here), but also because it merely involves configuring a messaging signal from the MMS User Agent to the MMS Relay/Server indicating the agent's availability to receive messages, and Shen already discloses what the contents of that message may be. Ex-1014, [0034].

706. As described above, TS-23.140-Shen renders claim 6 obvious.

2. ***Claim 12:*** *The networked system of claim 1, the link interface to encrypt messages identified for delivery to each given one of the wireless end-user devices to create secure Internet data messages, the device link agent on each given device further configured to decrypt the received secure Internet data messages for that device prior to delivering those messages to a respective software process.*

707. TS-23.140-Shen renders obvious a link interface (comprising the MM1 Transport Protocol and TCP/IP and transport layer security (TLS)-based transport protocols) that encrypts messages identified for delivery to wireless devices to create

a secure Internet data message (e.g., encrypted MMS messages), the device link agent on each wireless device configured to receive the message and decrypt it prior to delivering the messages to their respective software processes.

708. As I explained previously in 1[b1] (and incorporate here), TS-23.140 discloses delivering messages using MM1 Transport Protocols with TCP/IP and TLS protocols as a "link interface" (as claimed) for a TLS link between the MMS Relay/Server and UE/mobile device's User Agent. Although TS-23.140 discloses encrypting the connection between the push server and push client, it does not appear to disclose encrypting the messages themselves.

709. Shen, however, explains that the store and forward functionality of the MMS Relay/Server—without encrypting the messages—exposes a "potential security problem" given that such systems are "not an end-to-end solution …." Ex-1014, [0004].

710. To overcome this "security problem," Shen provides "an authentication and key management module 502 is added in the MMS gateway to distribute symmetric keys to users and a certificate management module 504 is added in the user device," as shown below:

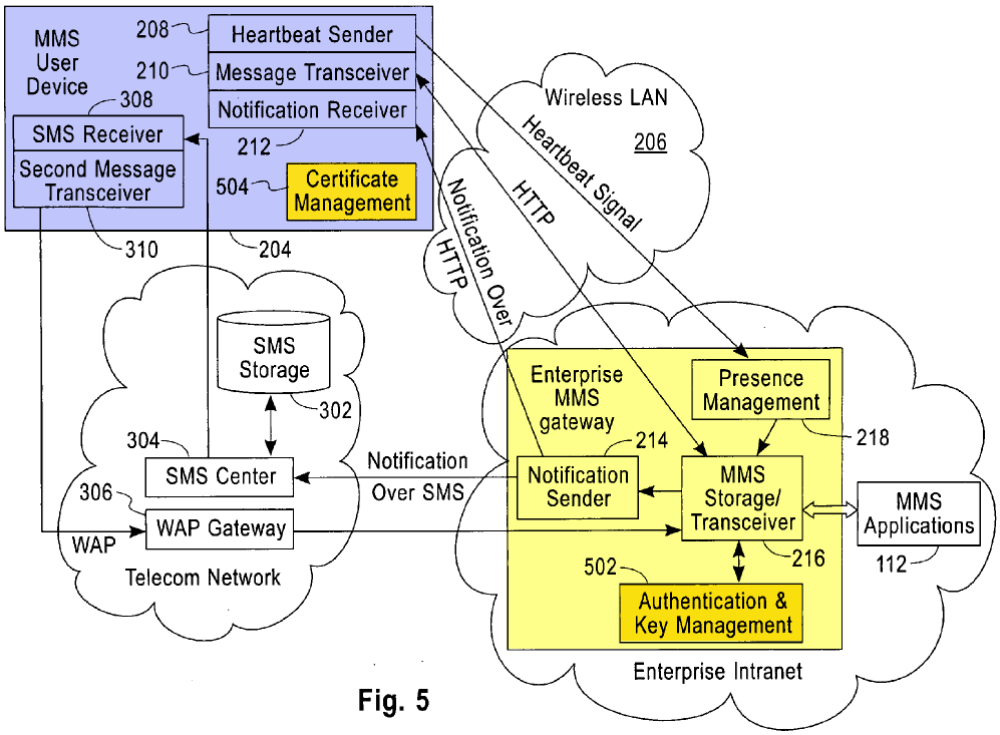


Fig. 5

Id., Fig. 5.

711. Using these modules, Shen discloses generation of a symmetric key that is then used by the MMS Relay/Server to encrypt data communications transmitted to the MMS User Device/Agent of user devices. Ex-1014, [0055]-[0060]; *see also id.*, [0029]-[0034]. Shen also discloses that messages can be "encode[d] and decode[d] to protect the privacy of these messages." *Id.*, [0054]

712. In my opinion, a POSITA would have been motivated to combine the teachings of TS-23.140 and Shen to achieve an MMS system "having improved security," resulting from "provi[sion] [of] an end-to-end security solution for MMS applications," which is particularly beneficial in the context of "enterprise

applications." *Id.*, [0017], [0021] (describing additional advantages stemming from Shen's invention). Specifically, a POSITA would have been motivated to implement an "authentication and key management module 502" to the stack including the MM1 Transfer Protocol with TCP/IP and TLS (*link interface*) that generates and "distribute[s] symmetric keys to users" and using this symmetric key to encrypt data transmitted to "the user device," as contemplated by Shen. *Id.*, [0054]-[0060].

713. Moreover, it was well known before the Critical Date for push servers to encrypt content of messages sent to push clients/client devices. Chou is one example of a push message system that uses a "WAP gateway" (e.g., a network message server) that encrypts content received from a "content server" for transmission to a "mobile client" that then "decodes the response." Ex-1009, 6:15-46. Rakic and Shen provide additional examples of a push server encrypting message traffic. Ex-1015, [0065]-[0068]; Ex-1014, [0054]-[0060]

714. Combining the above-described teaching of Shen with those of TS-23.140 would have been obvious to a POSITA at least because such a combination would have merely involved combining prior art elements (an MMS network and message encryption) according to known methods (using encryption schemes and key generators) to yield predictable results (encrypted messages). Similarly, the combination would have amounted to using a known technique (message

encryption) to improve similar devices (MMS networks) in the same way (securing content).

715. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification because Shen already envisions its proposal working in coordination with MMS-system architecture like that described in TS-23.140. Ex-1014, [0017]. Moreover, both TS-23.140 and Shen describe similar MMS environments and communications between MMS User Agents and the MMS Relay/Server, Ex-1004, pp. 16-18, 54-56; Ex-1014, [0029]-[0034], Figs. 1-2, 5, such that a POSITA would have found straightforward to modify TS-23.140's MMS Relay/Server to implement Shen's security modules.

716. Additionally, the resulting system's elements would each perform functions they were known to perform prior to the combination—i.e., TS-23.140's system would continue to facilitate communication of messages between MMS Relay/Server and MMS User Agents, and Shen's teachings (as implemented in combination) would facilitate encryption of messages by the MMS Relay/Server that are then sent to the MMS User Agent.

717. In my opinion, upon making the combination, a POSITA would have understood that TS-23.140-Shen provides additional encryption protocols to encrypt messages identified for delivery supplied to the components implementing the MM1 Protocol for delivery on the TLS link to encrypt messages, and thus create "secure

internet data messages." Although I previously noted in 1[e3], that use of TLS security protocols forms "secure internet data messages," a POSITA would have appreciated that messages can be "secured" multiple ways, including more than one way simultaneously. In this implementation, for instance, a POSITA would have appreciated that not only is the message encrypted (rendering a message a "secure internet data message") but so is the entire link between, e.g., a MMS User Agent and Relay/Server (also rendering the message a "secure internet data message"), meaning two layers of encryption would be applied in different parts of the stack of protocols.

718. And because it would have been obvious to provide encrypted messages, a POSITA would have further understood that each MMS User Agent would be configured to decrypt the received messages prior to delivering those messages to their respective processes (e.g., either for display to a user, or to a registered application). If decryption were not provided at the receiver, any received (encrypted) messages would be unusable by the user/applications receiving them because encryption involves scrambling data into an unrecognizable form.

719. For the above reasons, TS-23.140-Shen renders obvious claim 12.

3. ***Claim 13:*** *The networked system of claim 12, wherein the encrypted messages are transported to the device link agent using one or more of encryption on a transport services stack, IP (Internet Protocol) layer encryption, and tunneling.*

720. TS-23.140-Shen renders obvious that the encrypted messages are transported to the device link agent on a wireless device using encryption on at least the transport services stack.

721. As I explained previously in 1[b1] (and incorporate here), TS-23.140 discloses transmitting messages to the MMS User Agent using the MM1 Transfer Protocol with TCP/IP and TLS protocols, in which TLS provides encryption as part of a transport stack. Moreover, as I explained in claim 12 (and incorporate here), the combination of TS-23.140 and Shen teaches encrypting transmitted messages to/from the MMS User Agent.

722. For the above reasons, TS-23.140-Shen renders obvious claim 13.

4. ***Claim 16:*** *The networked system of claim 15, the device link agent on the given device further configured to buffer one or more of the upload messages for transmission to the network message server at a time selected by a heartbeat mechanism.*

723. TS-23.140 combined with Shen teaches or renders obvious that the device link agent (e.g., MMS User Agent) on a wireless device is configured to buffer upload messages for transmission to the network message server (e.g., MMS Relay/Server) at a time selected by a heartbeat mechanism.

724. First, TS-23.140 notes that terminals may "stor[e] MMs." Ex-1004, 19. A POSITA would have therefore understood that terminals have the capacity to "buffer" messages prior to transmission, since they are provided a place of storage.

725. Second, as I explained above in claim 6 (which I incorporate here), the combination of TS-23.140 and Shen teaches using a heartbeat mechanism for the client "to determine whether a wireless LAN is available." Ex-1014, [0033]-[0034]. In other words, Shen teaches a client device—like TS-23.140's User Agent—tests a network connection to a server by sending a heartbeat message.

726. In my opinion, upon combining TS-23.140 and Shen, a POSITA would have been motivated to configure the User Agent to buffer messages until the heartbeat mechanism determined it could reach the Relay/Server. It was well known that heartbeat messages typically required far less data than MMs—heartbeat messages could be on the order of bytes, *see, e.g.*, Ex-1043, [0062] (providing an example "64-byte link heartbeat (HB) message"), as opposed to kilobytes or even megabytes for MMs (to transmit multimedia such as images and video). A POSITA would have therefore been motivated to send a smaller amount of data to confirm the connection is still live *before* attempting transmission of the larger MMs. *See, e.g.*, Ex-1044, cl. 14 (describing "receiving data from the mobile device in the network, receipt of data from the mobile device occurring only after receipt of the connection confirm message in the mobile device.").

727. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification because TS-23.140 and Shen already provided the necessary software elements to carry out the modification—including a heartbeat

mechanism for sending heartbeat messages—and would simply require adjusting timers to trigger message delivery after the heartbeat confirmed the reachability of the Relay/Server.

E. Ground 1E: Claims 8 and 9 are Rendered Obvious by TS-23.140 and Pazhyannur

*1. **Claims 8-9: [8]** The networked system of claim 1, further comprising a secure server to store a secure authorization list, the secure authorization list indicating the authorized software components and the authorized network elements that are allowed to communicate using the network server system. **[9]** The networked system of claim 8, further comprising the respective device link agent on each wireless end-user device receiving access authorization information from the secure server, the access authorization indicating, respectively for each wireless end-user device, the software components authorized to receive messages via the device link agent on that device.*

728. TS-23.140-Pazhyannur renders obvious a secure server to store a secure authorization list to indicate authorized software elements (e.g., applications) and network elements (e.g., MMS User Agents resident on MMS client devices, MMS VAS Applications) that are allowed to communicate (e.g., registered) using the network server system (e.g., MMS Relay/Server), where each device link agent (e.g., MMS User Agent) receives access authorization information from the secure server indicating which software components (e.g., applications) are authorized to receive messages via the device link agent (e.g., MMS User Agent) on that device.

729. As I explained previously in 1[b2] and 1[c2] (and incorporate here), TS-23.140 discloses that applications must register with the User Agent to utilize

the MMS network. Moreover, TS-23.140 discloses that the Relay/Server interacts with a User Database that hosts "information for the control of access to the MMS." Ex-1004, 22. However, TS-23.140 does not appear to disclose the User Database maintains a list of registered network elements and applications.

730. Other prior art such as Pazhyannur discloses a network of user terminals and application servers. Ex-1037, Abstract. Pazhyannur recognizes that users may register multiple devices associated with the user, and the system maintains a centralized user profile that lists all registered devices and applications to permit access to the applications on any of the user's registered devices. *Id.*, [0040].

731. In my opinion, a POSITA would have been motivated to implement a list of registered network elements (such as user UEs/mobile devices, application servers, etc.) and applications in, for instance, TS-23.140's User Database to account for all devices and applications associated with a user (*id.*, [0040]), to "control of access to the MMS," which is functionality already assigned to the User Database, (Ex-1004, 22). In fact, it was well-known in the art to use lists of registered/authorized software/hardware elements to improve network security. *See, e.g.*, Ex-1008, [0044] ("the push authorization module 326 performs authorization of pushing a message to the client 310 based on a server whitelist which lists servers that are authorized to send a message to the client 310 through the gateway 320."), [0047].

732. In my opinion, a POSITA would have had a reasonable expectation of success in implementing a centralized list of registered network elements and applications because TS-23.140 already discloses that the User Database keeps a log of access control to the MMS network, and maintaining a centralized list of registered hardware and software simply requires compiling data the User Agents and Relay/Servers would already have, such as the identifiers for registered applications or network-hardware connecting to the Relay/Server.

733. Upon making this modification, in my opinion, a POSITA would have appreciated that User Agents and/or Relay/Servers would evaluate a list of registered network elements and applications in the User Database rather than a local list. That way, connections to the Relay/Server would be limited to already-approved hardware on the list, and applications would be manageable across different user devices.

F. Ground 1F: Claim 14 is Rendered Obvious by TS-23.140 and Ellison

1. ***Claim 14:*** *The networked system of claim 1, wherein the device link agent on a given device executes in a secure execution environment, and at least one of the software components on that device executes outside of the secure execution environment.*

734. TS-23.140-Ellison renders obvious executing the device link agent (e.g., MMS User Agent) in a secure execution environment, with applications that

interact with the device link agent operated outside of the secure execution environment.

735. TS-23.140 discloses both an MMS User Agent and applications executing on a UE/mobile device. Ex-1004, 17-18, 23. TS-23.140 does not, however, appear to disclose implementing the MMS User Agent in a secure environment.

736. Ellison is just one example of prior art that uses secure environments to inhibit attacks from malicious software. Ex-1019, Abstract, 1:16-51. To inhibit attacks, Ellison discloses both an "isolated execution mode" (depicted in red below) where "access … is restricted" and a "normal execution mode" that "operates in a non-secure environment, or a normal environment" (depicted in green below) "without the security features provided by the isolated execution mode." Ex-1019, 4:65-5:1, 6:1-26, 8:25-32, Figs. 1A, 1C. Additionally, Ellison's teachings enable "normal" execution, for instance, for applications "***outside of the secure execution environment on that device***." *Id.*, 4:65-5:1, 6:1-26, Figs. 1A, 1C.

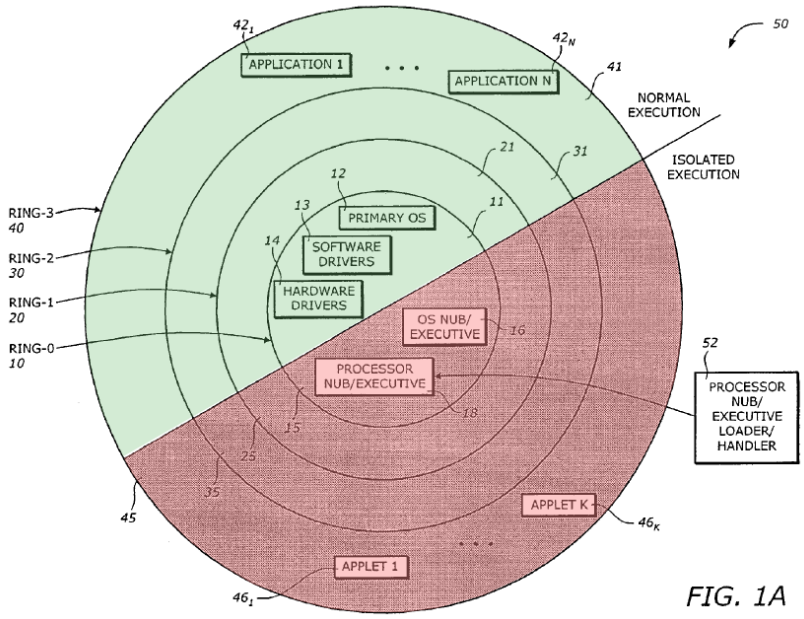


FIG. 1A

Id., Fig. 1A.

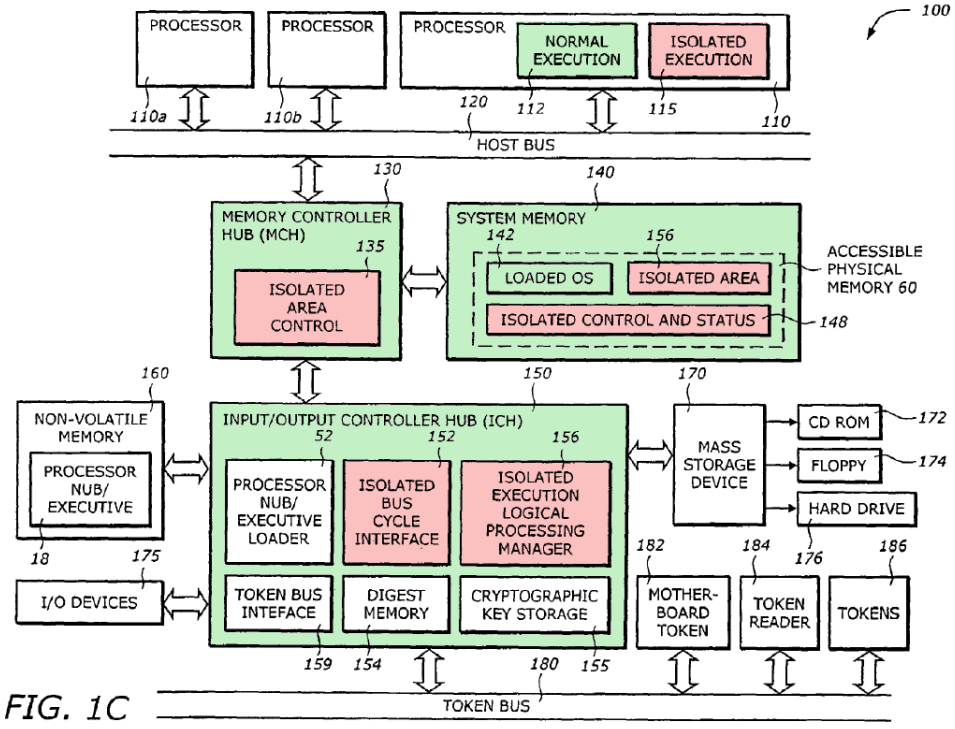


FIG. 1C

Id., Fig. 1C.

737. Ellison further discloses an "isolated area 70" which is "[a] memory area that is defined by the processor 110 when operating in [an] isolated execution mode." *Id.*, 6:1-26, 8:25-32, Figs. 1A-1C, 2. Ellison refers to this technique, illustrated below in Figure 2, as a "secure platform." *Id.*, 6:1-26, 8:25-32, 8:66-9:62, Figs. 1A-1C, 2.

738. Ellison further teaches that the secure platform includes a "key generator 240" that "generates a key operating system nub key (OSNK) 203" which is then supplied to "trusted agents." *Id*., 8:66-9:6, Fig. 2. One example of a trusted agent is a "usage protector 250" that "uses the OSNK 203 to protect the usage of [a] subset 230." *Id*., 9:28-40, Fig. 2. Ellison's usage protector 250 uses a hashing function with the subset 203's ONSK 203 to determine if the subset 203 has been altered following changes (e.g., reads or writes to the subset 203), thereby providing "protection against unauthorized reads, and detection of intrusion, tampering or unauthorized modification." *Id*., 9:47-62.

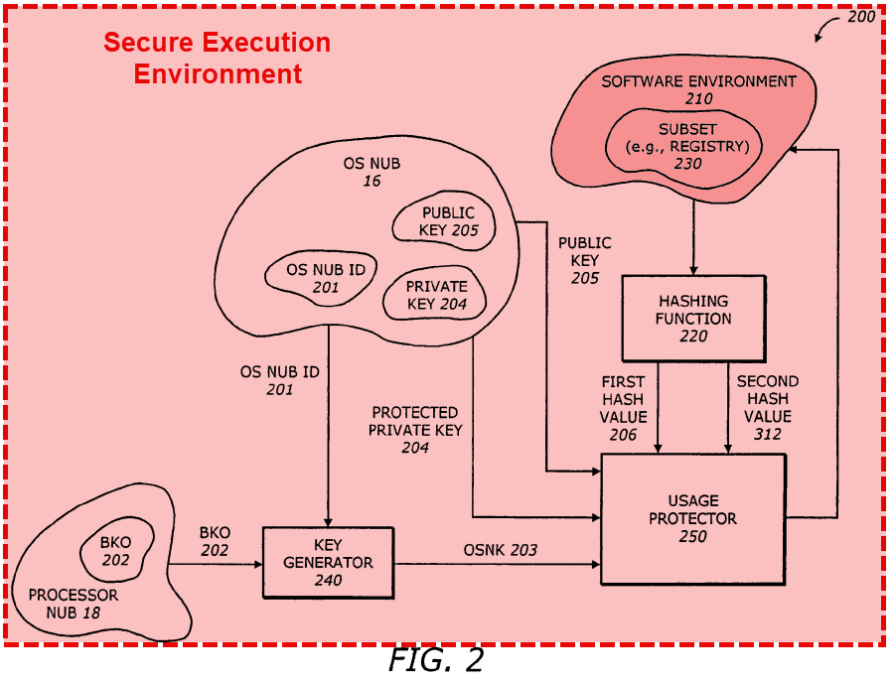


FIG. 2

Id., Fig. 2.

739. In my opinion, a POSITA would have found it obvious to combine these references such that Ellison's secure platform, which features hierarchical, nub-based normal and isolated execution environments, would have been incorporated into TS-23.140's UE/terminals. *Id.*, 8:25-32, 8:66-9:62, Fig. 2. A POSITA would have combined the teachings of these references for the following reasons.

740. First, a POSITA would have recognized that Ellison's secure platform would have improved the security of TS-23.140's MMS system by improving device security. *Id.*, 8:25-32, 8:66-9:62, Fig. 2.

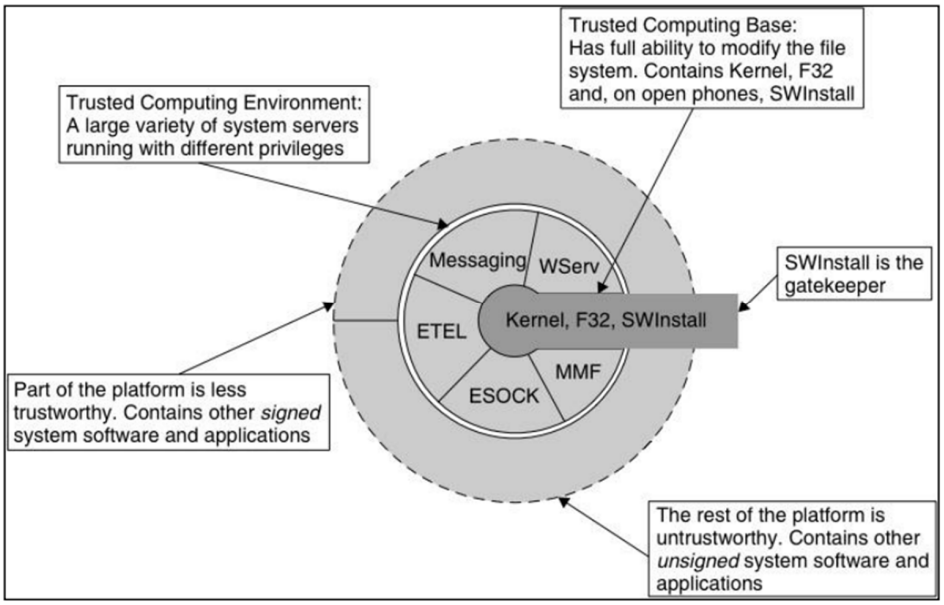
741. Second and relatedly, such increased confidence in device and/or network security provided by Ellison would have enabled a POSITA to TS-23.140's MMS services to additional UEs/servers (VAS Applications). *Id.*

742. Indeed, a POSITA would have understood that device level security improvements are an efficient, beneficial option for improving network security, which would have been a natural solution for TS-23.140's system, which contemplates secure/encrypted messaging in the MMS environments.

743. As explained below, combining the teachings of TS-23.140 and Ellison would have been obvious because such a combination would have merely involved combining prior art elements according to known methods to yield predictable results. Incorporating Ellison's security techniques into TS-23.140's mobile terminals would have been predictable and foreseeable with a reasonable expectation of success because Ellison describes that its techniques can be implemented in "computer system[s]" (such as TS-23.140's UEs) that would include a "processor." *Id.*, 5:11-16. Indeed, a POSITA would have recognized or found it obvious that UEs/terminals in TS-23.140 would include a "processor" capable of implementing Ellison's techniques.

744. Third, in the context of mobile devices and their associated operating systems, it was well-known before the Critical Date to implement more secure systems, with rings or tiers of protection around different device components. For

example, in the early 2000s, the Symbian mobile operating system (B. Li, et al., *Symbian OS platform security model*, LOGIN: Magazine (Aug. 2010), Ex-1033; Simon Higginson, Platform Security Concepts, in SYMBIAN OS PLATFORM SECURITY: SOFTWARE DEVELOPMENT USING THE SYMBIAN OS SECURITY ARCHITECTURE, 17-41 (Craig Heath ed., 2006)—Ex-1045) separated processes into three "Trust Tiers": The Trusted Computing Base (TCB), The Trusted Computing Environment (TCE), and Applications, as illustrated below:



Ex-1045, 24 (Fig. 1)

745. The first two layers/tiers shown were primarily dedicated to components of the Operating System with relatively higher operating privileged, followed by the Applications tier with applications having relatively lower privileges. The first two tiers included the Trusted Computing Base and the Trusted Computing Environment. As shown, the Trusted Computing Environment includes,

among other services, a Messaging service. The Application Tier could be further subdivided into Signed and Unsigned applications. Signed applications could interact with the TCE and have this layer execute higher privileged requests, whereas Unsigned applications could not access that same functionality.

746. Thus, it was well-known for operating systems to include the messaging services and clients/agents responsible for provision of those services, within the trusted computing environment, and for applications (including those that would be intended recipients of those messages) to be tiered/organized outside this environment. Given that TS-23.140 contemplates mobile terminals with an MMS User Agent that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging) from other applications/components of the environment, and Ellison's own disclosures confirming the same, a POSITA would have been motivated to implement Ellison's teaching of secure execution environment that enable such improved security within the MMS environment, such that the MMS User Agent is organized as part of the trusted computing environment (i.e., the secure execution environment) and enables communications with applications residing outside this trusted/secure environment.

747. In an example of the combined system, Ellison's secure platform would have been incorporated into TS-23.140's mobile terminals, to secure the MMS User

Agent, and further have a usage protector protecting messaging operations of the MMS User Agent. Ex-1019, 8:25-32, 8:66-9:62, Fig. 2. Specifically, Ellison's above-described teachings of a usage protector and secure platform, which operate within the secure, isolated area (secure execution environment) would be incorporated into TS-23.140's MMS User Agent executing on UEs/terminals, which in turn would provide secure access to applications residing outside the secure execution environment. Upon making this combination, a POSITA would have understood that TS-23.140's MMS User Agent would operate in an isolated, secure environment that "is protected by both the processor and chipset" (*id.*, 3:32-48), to reduce the MMS User Agent/user's messages exposure to attack. Applications that interact with the MMS User Agent would operate in normal or separate isolated environments to restrict the application(s) access to the MMS User Agent. *Id.*, 2:55-61 (noting applications operate in "Ring-3"—"the lowest level of the hierarchy" with the "least privilege").

748. A POSITA would have been motivated to so combine these teachings because (1) a POSITA would have recognized that Ellison's secure platform would have improved the security of TS-23.140's MMS system by improving device security (*id.*, 8:25-32, 8:66-9:62, Fig. 2); and (2) such device level security improvements are an efficient, beneficial option for improving network security,

which would have been a natural solution for TS-23.140's system, which contemplates secure/encrypted messaging in the MMS environments. *Id*.

749. In other words, the combined system would enable more secure transmission of MMS messages to applications on the mobile terminal, and prevent potential bad actors from sending malware to the mobile terminal in the form of such messages or from modifying or otherwise gaining unauthorized access to the messages. *Id.*, 8:25-32, 8:66-9:62, Fig. 2.

750. Moreover, combining TS-23.140 and Ellison's teachings would have been obvious because such a combination would have merely involved combining prior art elements according to known methods to yield predictable results. Incorporating Ellison's security techniques into TS-23.140's mobile terminals would also have been predictable and foreseeable with a reasonable expectation of success because Ellison describes that its techniques can be implemented in "computer system[s]" (such as TS-23.140's UEs) that would include a "processor." *Id.*, 5:11-16. Indeed, a POSITA would have recognized or found it obvious that UEs/terminals in TS-23.140 would include a "processor" capable of implementing Ellison's techniques.

751. Moreover, as noted above, it was well-known before the Critical Date for operating systems to include the messaging services and clients/agents responsible for provisions of those services, within the trusted computing

environment, and for applications (including those that would be intended recipients of those messages) to be tiered/organized outside this environment.

752. Given that TS-23.140 contemplates mobile terminals with an MMS User Agent that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging) from other applications/components of the environment, and Ellison's own disclosures confirming the same and providing such tiered, security architecture within the device, a POSITA would have been motivated to similar enable such improved security within the MMS environment by implementing the MMS User Agent as part of the trusted computing environment (i.e., the secure execution environment) and providing the applications that communicate with this agent outside this trusted/secure environment.

753. For the above reasons, TS-23.140-Ellison renders obvious that the MMS User Agent executes in a secure execution environment and one or more of the applications to which push messages are routed executes outside of the secure execution environment.

G. Ground 1G: Claim 17 is Rendered Obvious by TS-23.140 and Fok

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

interprocess communication service.

754. TS-23.140-Fok renders obvious a secure interprocess communication service that is separately secured from the secure Internet data message link, with a device link agent (e.g., MMS User Agent) on a given device causing messages to be securely delivered to a software process (e.g., application process) by initiating delivery on the secure interprocess communication device.

755. As discussed above in 1[b2] (and incorporated by reference here) TS-23.140 discloses the transmission of application-specific MMs through the MMS network. However, because TS-23.140 is a standards specification concerned with *network* interactions, it does not expressly describe inter-application specifications on, e.g., a given device.

756. Fok describes a mobile environment that includes a secure inter-application communication platform. Ex-1038, [0036]-[0038], [0081], [0106]-[0108], Figs. 4, 8-9. Fok uses a "handshake" between a "primary application" and one or more "recipient" applications to confirm that they are trusted application using "a list of unique identifiers" before transmitting information between the two (causing messages to be securely delivered to a software process). *Id.*, [0047]-[0053], [0096]-[0102], [0106]-[0108], Figs. 7-10.

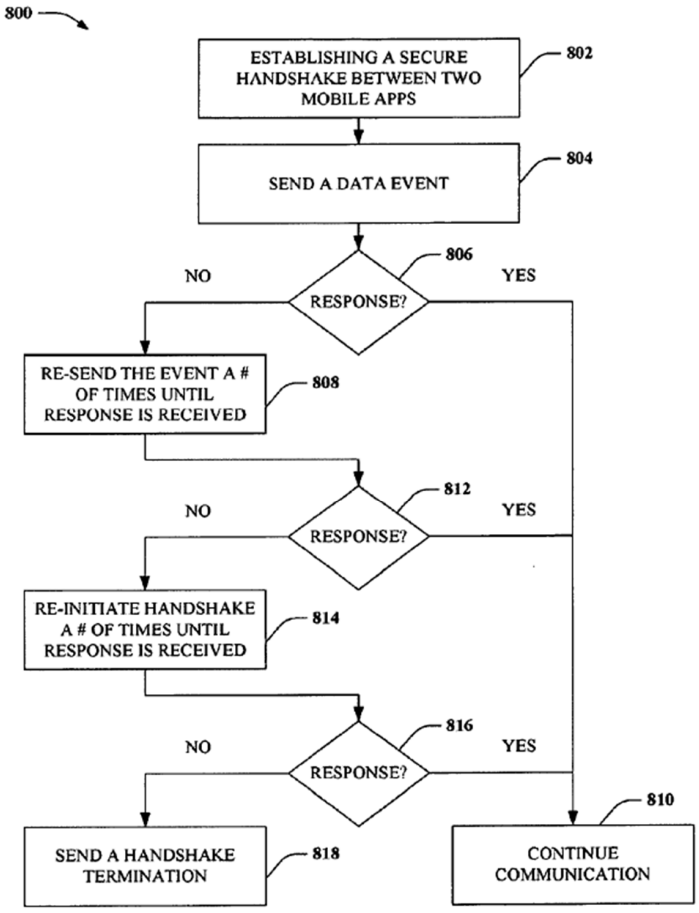


FIG. 8

Id., Fig. 8.

757. In my opinion, a POSITA would have been motivated to combine TS-23.140 and Fok to provide a "secure inter-application communication" like Fok's so that two or more applications residing on a single UE/wireless device may transmit application-specific MMs to one another through the MMS User Agent. This would effectively provide MMS-compliant applications to interact without needing to develop, for instance, custom APIs ("Application Programming Interfaces") for alternative forms of direct communication, and ensure the applications were trusted

before sharing information; the MMS User Agent would provide the interface and relevant authorizations. As one example, an MMS-compliant messaging application could be configured to forward user-specified MMs with images to a photo-storage or editing application on the same device. Or a messaging application could direct a video-playback application to download a specific codec to play back a video message in the video-playback application.

758. A POSITA would have understood that this inter-application communication would be secured separately from the link established using MM1 Transport Protocol and TCP/IP and TLS protocols because inter-application messaging does not need to utilize an internet network or reach the Relay/Server when transmitting messages between different applications on a single UE/wireless device (since both applications exist on a single device). A POSITA would have appreciated that this separate inter-application channel would also be "secured" to prevent unauthorized access to the message transmissions between the two applications. *See, e.g.*, *id.*, [0036]-[0038]. Moreover, although Fok discloses a direct handshake between the two applications, a POSITA would have appreciated that the combination of TS-23.140 and Fok would instead pass through the User Agent (so one application sends the User Agent a message, which the User Agent then forwards to the receiving application). That way, each application would not need to

be configured with a brand-new "handshake" mechanism, but would instead utilize its existing protocols for MMS transmissions.

759. Moreover, combining TS-23.140 and Fok's teachings would have been obvious because such a combination would have merely involved combining prior art elements according to known methods to yield predictable results. Incorporating Fok's inter-application communication techniques into TS-23.140's mobile terminals would also have been predictable and foreseeable with a reasonable expectation of success because Fok describes that its techniques are directed towards mobile operating environments, similar to TS-23.140's UEs and MMS User Agents, Fok, [0002].

760. In my opinion, a POSITA would have had a reasonable expectation of success in making this combination because Fok already describes how to establish an inter-application connection; the only difference is the connection would be established through the MMS User Agent. Moreover, this internal-facing, device-specific connection would only supplement, and not replace, TS-23140's external-facing, networked connection.

H. Ground 2A: Claims 1-7, 10-11, 15, and 18 Are Rendered Obvious by Houghton and Munson

1. Combination of Houghton and Munson

761. In my opinion, a POSITA would have found it obvious to combine the teachings of Houghton (Ex-1007) and Munson (Ex-1017), and specifically to

incorporate Munson's store-and-forward functionality into Houghton's push messaging system.

762. Specifically, in my opinion, a POSITA would have been motivated to incorporate computer memory and programming to deliver messages from memory—consistent with Munson's teachings—to improve flexibility and reliability of the system.

763. For instance, a POSITA would have recognized or found it obvious that a mobile terminal that is an intended recipient of a message may not have network connectivity at the time a message originator initiates delivery of a message. In this instance, the message would be stored in the computer memory until the mobile terminal reconnects to the network and the message can be delivered. Indeed, Houghton already contemplates such "store and forward messaging systems," noting that messages may not be deliverable while the user is out of network coverage. Ex-1007, 3, 7 ("These server computers such as a telecommunications operator SIP Server or WAP Gateway is used to control and deliver messages and ***store messages when the mobile terminal is unavailable and deliver them on the sender[']s behalf, possibly later then the terminal becomes available***"). Moreover, it was well-known for store-and-forward push systems to include buffers to store messages "when the mobile terminal is not connected to the mobile internet service network" and "re-transmitting the stored push information when the mobile terminal is connected to

the mobile internet service network." Ex-1006, [0034], claim 14, Fig. 1. A POSITA would have therefore been motivated to implement computer memory and programming to deliver messages from memory, consistent with Munson's message-buffering solution to avoid losing or discarding of messages (e.g., when the mobile terminal is outside network range or connectivity and the message is therefore undeliverable).

764. Moreover, a POSITA would have appreciated that, by providing computer memory to store messages mid-transmission, both the originator and recipient would be able to better control the conditions of message delivery. Houghton, for instance, discloses that the "push message from the push server 401 may be triggered by any trigger event, local or remote, ***defined at the server***. The trigger may, among others, include an alarm, notification, or measurement result received to the push server 401 from another device or system." Ex-1007, 21; *see id.*, 14 (describing various trigger events), 21-22, 25-28 (describing various trigger events). Munson similarly teaches messages can be delivered from memory "according to a schedule of time ***or event***" such as coordinating pushes "for a system maintenance purpose during off-peak hours." Ex-1017, [0040], [0044]. By providing memory along the transmission path to buffer messages, both the originator and recipient would be able to define additional conditions to complete delivery of the message without having to rely on, e.g., the availability/connectivity of the

originator/recipient devices once a delivery condition is triggered; they would be delivered from the intermediate memory instead. *See also, e.g.*, Ex-1004, 27-29 (describing both originator-defined and recipient-defined delivery conditions to deliver messages from a central memory buffer).

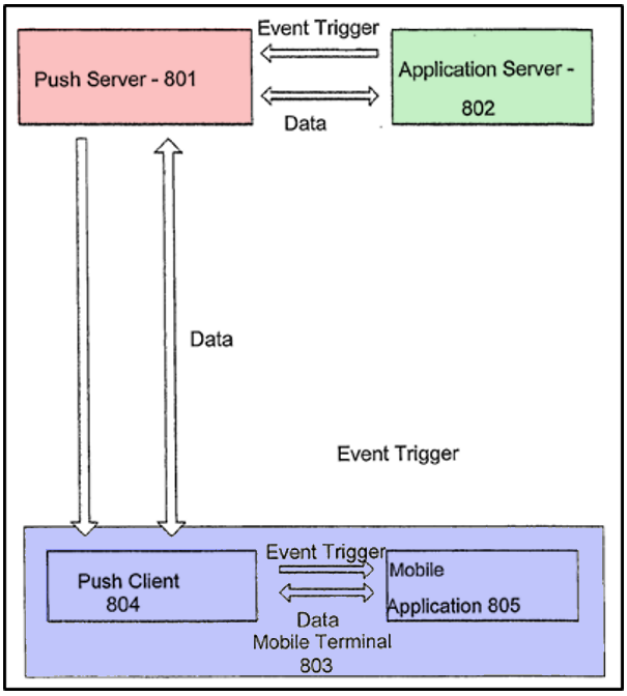
765. In my opinion, combining Houghton and Munson's teachings would have been obvious and straightforward because such a combination would have amounted to using known prior art techniques (message buffering) to improve similar devices/systems (push messaging systems, as contemplated by Houghton and Munson) in the same way that yields predictable results (push messaging systems that can store and forward messages and subsequently deliver them, as contemplated by Houghton and as expressly described by Munson).

766. A POSITA would have had a reasonable expectation of success in making this modification because Houghton already envisions not only store-and-forward systems that provide centralized computer memory to buffer messages, but also conditioning delivery upon the occurrence of certain events. Ex-1007, 3, 7, 14, 21-22, 25-28. All Munson provides is additional details about how to implement a store-and-forward system by providing the computer memory at a server, and the computer programming to deliver messages from memory (including upon occurrence of certain events). Ex-1017, [0036]-[0038], [0044], Fig. 4. The resulting combination would merely require generic computer memory (at a central server) to

serve as the message buffer, and programming to store incoming messages in memory and then deliver them from memory (including at the occurrence of a triggering event both Houghton and Munson already envision).

767. Thus, in the combination, each of Houghton and Munson's systems would continue to operate the way they did prior to combination, e.g., Houghton's system would continue to push messages between a push server and a push client, and Munson's described techniques (as implemented in combination) would facilitate buffering of push messages in a buffer/memory of Houghton's push server and delivering the stored messages when certain conditions arise (e.g., user becomes available in network range).

768. In an example of the resulting combination (referred to as Houghton-Munson) and as illustrated below, Munson's handler units and buffer (which handle and store messages to ensure delivery) and content senders (which coordinate message transmissions) would have been incorporated into Houghton's push server such that they would buffer push messages based on the policies (as further described by Munson) and subsequently deliver those messages upon occurrence of certain conditions. *Id.*, 3, 7, 11-14, 21-23, Figs. 3-4, 8; Ex-1017, [0036]-[0038], Fig. 3-4.



Ex-1007, Fig. 8 (modified to incorporate Munson).

769. Finally, in my opinion, a POSITA would have been further motivated and had a reasonable expectation of success in implementing some of the additional delivery conditions Munson taught in Houghton's system—to the extent Houghton does not already disclose the same triggers as Munson—to provide additional flexibility and conditions to control delivery. Houghton, for instance, discloses that message delivery may be conditioned upon "***any trigger event***, local or remote," Ex-1007, 21, and therefore already contemplates additional conditions may be incorporated into the system. All Munson provides are some additional or alternative "trigger events." Ex-1017, [0040], [0044]. Implementing additional trigger events in Houghton's system would have required only some computer programming at the

server, which, in my opinion, would have been well within the skills of a POSITA to implement.

2. Claim Analysis

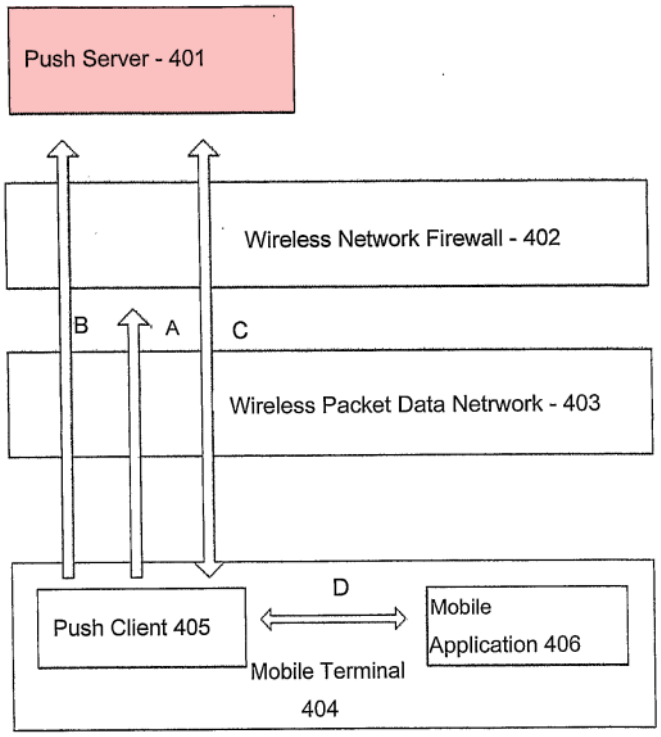
a. Claim 1

i. 1[pre]-1[a]: A networked system comprising: a network server system including

770. If the preamble is limiting, Houghton discloses or renders obvious a networked system (e.g., a network of push clients and push servers) comprising a network server system (e.g., push server).

771. In detail, Houghton discloses a network of servers, clients, and computing devices that includes a push server (401/801) to send push messages through the network. Ex-1007, Abstract, 16-17, 21 (noting system may include "other servers, mobile terminals, push clients and computing devices"), Figs. 4, 8.[8]

[8] Houghton provides several figures and/or embodiments that use different reference numbers for the same structures (e.g., "push server" is provided reference number 401, 501, 701, 801, 901). Although Houghton describes these as different embodiments, a POSITA would have understood that, for the most part, their teachings are not mutually exclusive of one another, but instead describe different aspects or procedures, sometimes involving additional network elements, that a single system may be configured to carry out. *Compare, e.g.*, Ex-1007, 15-16 (noting Figure 4 describes one "mobile terminal" embodiment of the network system, Figure 6 depicts a process to maintain an open network path across the network, and Figures 7-9 depict different message-push procedures that incorporate an additional "application server" into the network), *with id.* (noting Figure 5 describes an *alternative* wired terminal embodiment). For this reason, I often cite to reference numbers from Figure 4, but I also rely upon other teachings provided in the context



Ex-1007, 15, Fig. 4 ("illustrating an exemplary implementation of the system for pushing mobile application launch commands to a mobile terminal based on server side events in accordance with one embodiment of the present invention").

772. "Figure 4 illustrates a Push Client basic functionality 405 implemented in a mobile terminal 404 ***as part of a message passing system in cooperation with a Push Server 401***. The push client 405 is preferably implemented in the form of a software run in a processor of the mobile terminal 404. The push client software 405 runs in the background and attempts to log into the push server 401 over a data

---

of other figures (e.g., Figures 6-8), because the features associated with these other "embodiments" may obviously be implemented together.

network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets." *Id.*, 16-17; *see id.*, Abstract ("A server outside a wireless network security entity is adapted to push messages to a mobile terminal located in the wireless network and thereby cause programs to start to a specified operating state or, if already started, change operating state to the state indicated by the message thereby presenting a new application view to the user of the mobile terminal. ***This mobile-initiated permanent connection allows the server to pass or push messages from outside the operator's firewall to the mobile terminal at any time.***"); Figs. 4, 8. The system in Houghton may include "other servers, mobile terminals, push clients and computing devices." *Id.*, 21.

773. Additional details regarding Houghton-Munson's network is described with reference to 1[b1]-1[e4] (which I incorporate here).

774. For the above reasons, Houghton-Munson renders obvious 1[pre] and 1[a].

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

775. The '320 patent discloses use of transport protocols for such server-client communications, including, e.g., TCP, IP, UDP (Ex-1001 ('320 Patent), 99:8-32) and notes that "there are many other approaches for implementing a secure control channel over the Internet including, for example, one or more of various

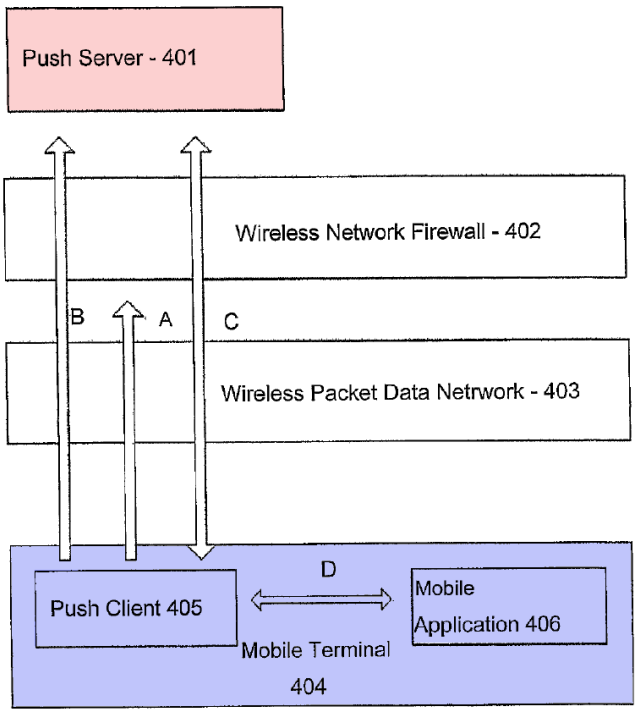
packet encryption protocols running at or just below the application layer, running TCP Transport Layer Security (TLS), and running IP level security or secure tunnels" (*id.*, 17:13-22).

776. Consistent with these disclosures, Houghton discloses or renders obvious a link interface that maintains a secure message link through an Internet network between the message link server (e.g., push server 401) and a device link agent (e.g., push client 405) on respective client devices (e.g., mobile terminals 404). Houghton discloses that 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"). Ex-1007, 17, Fig. 4 (below; annotated).

777. For example, and as explained below, Houghton discloses communications over the Internet network between the push server 401 and push client 405, using various transport protocols that have additional security to facilitate secure links between the push server 401 and the push client 405.

778. Specifically, Houghton's push client software 405 on mobile terminal 404 "runs in the background and attempts to log into the push server 401 over a data network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets. The arrow A in Figure 4 illustrates an optional attempt by the push client 405 to communicate with the wireless network firewall 402 or other similar

security or gateway device in order to open a connection for the push server 401 to directly contact the push client 40[5] as in FIG 3. If successful, operations can continue as described in FIG 3 or as below. Arrow B illustrates a login attempt." *Id.*, 17, Fig. 4 (annotated below; showing communication between push client 405 and push server 401 over a wireless network).



Id., Fig. 4.

779. Push client 405 constitutes the device link agent because it is "software" running on the mobile terminal device that "initiates and maintains" a link using "Internet technologies" between the mobile terminal 404 and the push server 401, and using this link, push server 401 pushes messages to the mobile terminal. *See, e.g.*, *id.*, 11, Abstract ("A server outside a wireless network security entity is adapted

to push messages to a mobile terminal located in the wireless network and thereby cause programs to start to a specified operating state or, if already started, change operating state to the state indicated by the message thereby presenting a new application view to the user of the mobile terminal. ***The mobile terminal is provided with means, such as software, which initiates and maintains contact with the server using Internet technologies***. This mobile-initiated permanent connection allows the server to pass or push messages from outside the operator's firewall to the mobile terminal at any time."), 19-20 (describing a data structure stored in mobile terminal including preferred communication protocol and other network connection parameters for facilitating connection between push client and push server), 20 ("At any time after a connection is established between the client 405 and the server 401, the client may push a message to the server or the server may send a push message to the client. This is illustrated by means of the arrow C in Figure 4. ***These messages through a previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS arrive quite rapidly***…. ***A connectionless protocol, such as UDP/IP, may also [be] used in cases where the connection-oriented protocol is not possible or not desirable because of the push server resources used by every static instance of a connection-oriented protocol***. Connectionless communication is also used when a message needs to get through to the server 401 quickly, such as when occurs when the client 405 notifies the server 401 of a[] terminal event, user interface

event or application event but the connection-oriented protocol connection is not connected or is already busy transferring information….”); Ex-1001 (’320 Patent), 43:18-50 (interchangeably using agent and software (such as an application) running on a device).

780. For this connection, Houghton contemplates use of known transport protocols (e.g., a link interface), including, e.g., “a connection-oriented protocol suitable to the wireless network such as TCP/IP,” “a connectionless protocol suitable to the wireless network 403, such as UDP/IP,” “an alternate connection-oriented protocol such as HTTP suitable for mobile push operation through more strict firewalls and proxy Internet connections,” or “a secure protocol such as HTTPS, IP-Sec, secure IP6 or a proprietary security protocol to identify the communicating parties, prevent message interception by 3rd parties, and prevent message modification by 3rd parties.” Ex-1007, 18-20; *see id.*, cl. 10 (“method as claimed in claim 9, wherein said trying of one or more other techniques and/network parameters include one or more of using an alternate push server network address which may be more available or more accessible at the present time and from the present location, using one or more alternative mobile terminal network connection settings, or variation of the parameters within the settings, using an alternate connection route through a second wireless or fixed terminal accessible to the wireless terminal over a wireless network and capable of communication with the push server, using a TCP

connection, using a UDP connection, using an SSL connection, using an HTTP or tunneling HTTP connection, using an SHTTP or tunneling HTTPS connection, using an XML-based connection over other transport protocol.").

781. "At any time after a connection is established between the client 405 and the server 401, the client may push a message to the server or the server may send a push message to the client," which is illustrated using arrow C in Figure 4. *Id.*, Fig. 4, 20 (explaining using of "a previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS" for communication between push client and push server). Thus, Houghton describes server-client communications using secure-transport protocols (secure Internet data message link) similar to those in the '320 specification. Ex-1001 ('320 Patent), 16:66-17:22, 39:20-32, 99:8-32, 100:21-28, (identifying secure encryption protocols, e.g., TLS/SSL, IP-Sec, HTTPS).

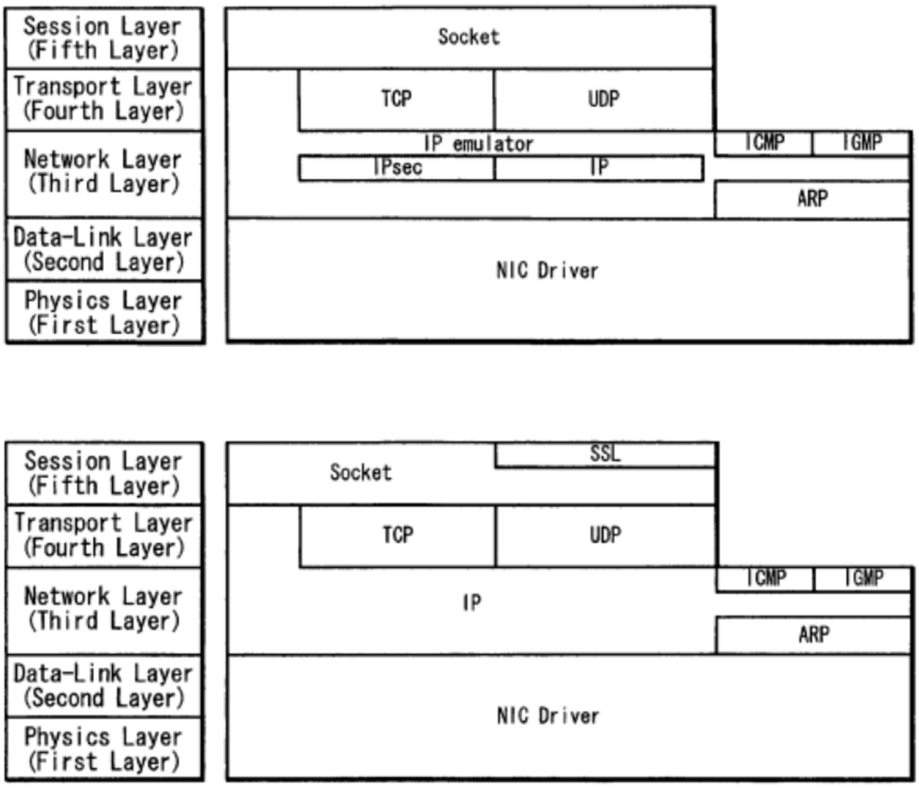
782. In some embodiments, the '320 patent describes the communication as being secured using encryption, TLS/SSL, HTTPS, IP-Sec, or another security mechanism. Ex-1001 ('320 Patent), 16:66-17:22, 39:20-32, 90:33-61, 99:8-32; 100:8-32. The above descriptions in Houghton similarly describe the server-client communications using such secure transport protocols. From these disclosures, a POSITA would have understood or found it obvious that Houghton's push server would include a "stack" of transport protocols to facilitate its secure communications

over an Internet network with the mobile terminal 404. Ex-1007, 1 (noting TCP and UDP protocols—which operate at the "transport" layer of the OSI model—"run on top" of "IP protocols") 17-20, Fig. 1.

783. Indeed, before the Critical Date, it was well-known for a device implementing transport protocols (as already contemplated by Houghton) to implement a transport services/protocol stack that facilitates secure network communications with other network elements (e.g., a mobile terminal).

784. For example, a prior art reference, Ozaki, describes the state of the art for TCP/IP protocol stacks that are implemented in a device's "software or firmware" and that are made up of multiple layers, including a physical layer, a data link layer, a network layer, a transport layer, a session layer, a presentation layer, and an application layer. Ex-1005, [0012]-[0014]. The physical and data link layers are typically implemented using a device's network interface card (NIC) and an associated driver, and enable connecting the device to the network. *Id.*, [0015]-[0018]. The network layer "defines a communication method between two … nodes" and in the "case of TCP/IP" this layer corresponds to an IP layer, and the transport layer corresponds to the TCP layer. *Id.*, [0019]-[0021]. IPsec (Internet Protocol Security) is used to encrypt communications in the network layer. *Id.*, [0019]-[0021], [0008]-[0010] (describing use of symmetric/common key encryption for IPsec encryption). The session layer "defines a procedure of a session (from start to end of

a communication) and enabling a connection between applications," and Secure Socket Layer (SSL) is "conventional representative encryption communication technology" for facilitating encryption of communication in this layer. *Id.*, [0022].



Ex-1005, Figs. 25-26.

785. Given that Houghton already contemplates a stack of TCP/IP-based transport protocols for facilitating transport of data/messages between network entities, a POSITA would have understood that Houghton would have implemented any well-known TCP/IP transport stacks in the software and hardware of the push server to enable such data transport over a network.

786. Moreover, as described above, a communication link is established between the push client/mobile terminal and the push server 401, and this link constitutes a secure Internet data message link because it facilitates sending of push messages between the push client 405 and push server 401 using secure/encryption protocols such as IPsec, HTTPS, and/or SSL. *Id.*, 11, 18-20, cl. 10; Ex-1001 ('320 Patent), 16:66-17:22, 39:20-32, 70:31-44, 90:33-61, 99:8-32, 100:8-32, (similarly describing use of TLS/SSL, IPSec to secure network communications between network entities).

787. Given these disclosures in Houghton about using secure protocols such as SSL, HTTPS, and IPSec, and the well-known use of such protocols to secure TCP/IP-based network communications, a POSITA would have understood or found it obvious that the link interface that would be implemented in Houghton's push server would use such protocols for securing the TCP/IP based communication link between the push client 405 and the push server 401.

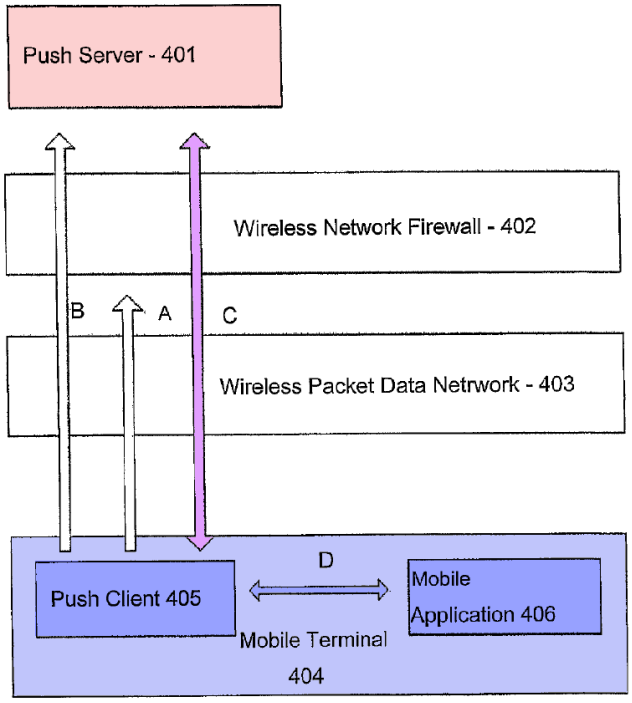
788. A POSITA would have further understood that, in push messaging environments such as those provided in Houghton, multiple mobile terminals (and their associated push clients) would be in communication with push server 401, such that secure SSL or IPSec-based TCP/IP communication links would be between the push server and each mobile terminal and its respective push client. *See also* Ex-1007, 18, 19 (disclosing multiple "terminals" in Houghton's environment).

789. For the above reasons, Houghton-Munson renders obvious 1[b1].

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

790. Houghton discloses or renders obvious that each of the wireless end-user devices (e.g., mobile terminals 404) include multiple software components (e.g., mobile applications 406) authorized to receive messages (e.g., application-specific messages) via a device link agent (e.g., push client) on that device.

791. As shown below and described further in Houghton, wireless end-user device (mobile terminal 404) includes multiple software components (e.g., mobile application(s) 406), and messages (e.g., command push or application command messages) are provided to the respective applications via a device link agent (e.g., push client 405) on the device/mobile terminal 404. Ex-1007, 21-22 (disclosing messaging that identifies "which mobile application 406 from a plurality of such applications…. [T]he push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406"), 24-25, 27, Figs. 4, 8.



EX-1007, Fig. 4.

792. Push client 405 is "implemented in the form of a software run in a processor of the mobile terminal 404" and operates "as part of a message passing system in cooperation with a Push Server 401." Ex-1007, 16-17; *see id.* ("The push client software 405 preferably launches automatically when the mobile terminal 404 is turned on. The push client software 405 runs in the background and attempts to log into the push server 401 over a data network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets. The arrow A in Figure 4 illustrates an optional attempt by the push client 405 to communicate with the wireless network firewall 402 or other similar security or gateway device in order to open a connection for the push server 401 to directly contact the push client 40[5] as

in Figure 3. If successful, operations can continue as described in Figure 3 or as below. Arrow B illustrates a login attempt."); *see also* Ex-1001 ('320 Patent), 43:18-31, 11:51-57 (describing the device agent as software executing in a device processor).

793. These messages are received over a secure TCP/IP link by push client (405) operating on mobile terminal (404) before being passed to the relevant application (406). Once a connection is established between push client 405 and push server 401, "the ***server may send a push message"*** to push client 405, as illustrated using arrow C above. Ex-1007, 20, Fig. 4. Such a push message is transmitted to the push client (for delivery to a mobile application) through the "previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS." *Id*.; *see id.*, 34, cl. 33 ("wherein ***the push client*** and push server establish a secure connection and ***unwrap[s] such security*** on behalf of the mobile application"); 16-17 (noting that the push client (405) operates in software); Ex-1001 ('320 Patent), 43:18-31 (explaining that device-link agents can be "implemented largely or entirely in software"). Because the communication link uses SSL to secure communications transmitted over the communication link between push server 401 and mobile terminal 404, the received push message from push server constitutes a secure message link message.

794. Additionally, a POSITA would have understood or found it obvious that push server 401 may communicate and facilitate transmission of push messages to multiple mobile terminals, each including its respective push client 405 and mobile applications 406. This is confirmed and corroborated by the following disclosure in Houghton: "The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 and direct such to the push server 401, possibly for redirection through push server API to other servers, ***mobile terminals, push clients and computing devices***." Ex-1007, 21. Therefore, the mobile applications 406 are configured to receive/process message data. *Id*.

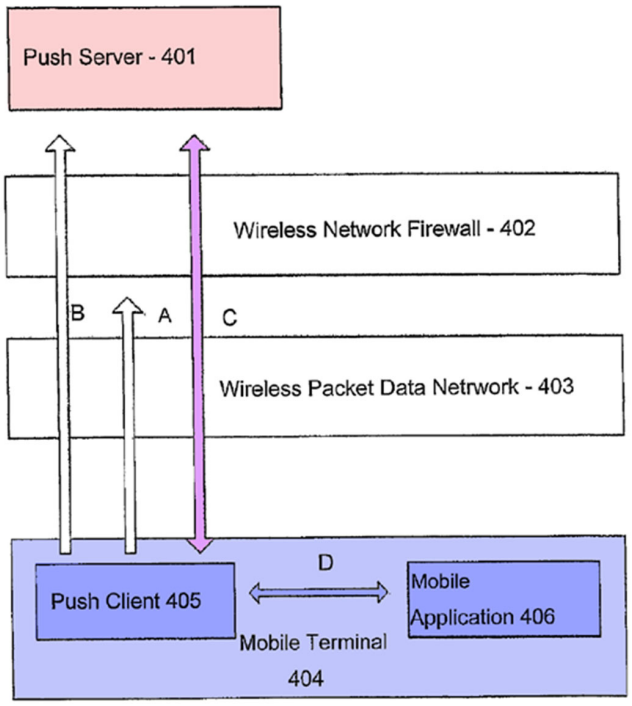
795. The received push message include "application-specific data," such as "application commands," "updates," or "data messages" that are "***directed to a push application or dynamically loaded module or user interface feature from among a plurality of such applications, modules and features by means of said client-initiated or client-opened and client-maintained permanent connection***." *Id.*, 8, cl. 1. "[T]he push messages may include, for example, a message stimulating, triggering or ***commanding the launch, display or shift to foreground of a user interface feature of the terminal such as an application*** which is part of the basic terminal software or and an additional installed application beyond the base software supplied with the mobile terminal." Ex-1007, 11.

796. In response, the "mobile terminal may spontaneously open the application, dynamically loaded module or user interface feature of the terminal thereof to a designed application view or location, present new information of the availability of new information, a new feature or operating state." *Id.*, 12. And, if already operating, "the action of such a message would instead stimulate the application or user interface feature to present itself to the foreground of the user interface and change state as specified in the application-specific message received from the server." *Id.*

797. The push message also includes "a data packet" with "binary or text information ***specifying which mobile application 406 from a plurality of such applications***" and additional "binary or text information to be passed to the … specified mobile application." Ex-1007, 21, 22 ("The potential applications of the COMMAND PUSH include but are not limited to mobile applications for enterprise connectivity, back office integration, database access, scientific and industrial process and equipment monitoring and control, pictures transfer, mobile gaming, application data transfer, mobile electronic commerce, application alarms, reminders, multimedia interactivity, corporate calendaring, personal messaging and group messaging, etc.").

798. The push message is subsequently directed from push client 405 to the specified mobile application 406 via the communication path shown using arrow D

in Figure 4 below. The figure also shows the communication relationship between the push server (401) and push client (405) using arrows B and C.



Id., Fig. 4 (annotated), 21, 24.

799. A POSITA would have understood or found it obvious that push server (401) does not establish a link with just one mobile terminal (404) with one push client (405), but instead communicates push messages to multiple "mobile terminals," each including its respective push client (405) and mobile applications (406); Ex-1007, 21 (routing messages to "servers, mobile terminals, push clients and computing devices").

800. Moreover, because the push command messages are specifically routed to a particular application from among a plurality of applications running on the

terminal, and the destination application then operates on the received message data (e.g., launch application, present content, etc.), a POSITA would have understood or found it obvious that the applications executing on the mobile terminal are authorized to receive and process the data included in the command push/application command messages.

801. Additionally, or alternatively, even if it is determined that Houghton does not explicitly disclose that the multiple software components (e.g., mobile applications 406) are ***authorized*** to receive and process data from secure message link messages, a POSITA would have found that to be obvious in view of the relevant state of the art, as explained below.

802. Before the Critical Date, it was well-known for applications that are intended recipients of push messages to register with a push server and/or push client before they can receive messages transmitted over such push frameworks. For example, in the relevant context of push communications and push environments, Lee discloses that "various push services provided to ***each of [the] applications are managed*** in an integrated fashion through the integrated push service server 100 and the integrated push service agent 250" and for this reason, specification of a recipient application (similar to Houghton's disclosures) are required in push messages to facilitate appropriate transmission of the push message to the application. Ex-1006, [0023].

803. Similarly, Shenfield discloses another push environment in which push client includes "an application registration application provider interface adapted to register applications to said push client and further adapted to receive an application manifest for said applications." Ex-1018, [0017] (explaining that the push client may "register with only one application"), [0063], [0106]-[0109] (application "registers with push client 510, providing application manifest 918 to establish a channel to a content provider for servicing" the application); Ex-1004, 54-56 (explaining that "[a]pplications that want to transport data specific to applications other than MMS will initially need to register with the appropriate MMS User Agent or MMS VAS Application").

804. A POSITA would have found it obvious to implement the above-described and well-known authorization or registration processes in Houghton's system to enable registration of a particular application with a push client and the push server with which the push client interacts.

805. A POSITA would have been motivated to do so for multiple reasons, including to ensure the application is compatible with the push client/server's communication protocols and to enable dynamic content delivery "to have information or data pushed" to devices without users of such devices having to "go and seek out that data." *See, e.g.*, Ex-1018, [0003]-[0006].

806. A POSITA would have been further motivated to implement application registration in Houghton's system to ensure that data and messages are routed to the correct application and simultaneously ensure that applications have opted into receiving push data/messages (thus avoiding inundating unregistered applications with content that they have not requested or opted into receiving). As an example, different applications likely expect incoming data in a variety of formats. Whereas weather data may come in one format (a series of numbers to be parsed and organized by the application), news applications are likely to send full text sentences to be read. Neither format in this example would make sense to the other application, and therefore delivering messages to them would result in confusion. As such, it is clear that a POSITA would have been motivated to ensure that information was routed to the correct application.

807. Moreover, because Houghton already contemplates that the push client and push server coordinate on message delivery to and from applications resident on mobile terminals, implementing the well-known application registration/management teachings in push environments, within Houghton's similar push environment, would have been straightforward and a POSITA would have had a reasonable expectation of success in combining the teachings in the above-described manner.

808. Therefore, a POSITA would have found it obvious that the push messages in Houghton are routed by push client 405 to applications that are authorized/registered to receive such messages.

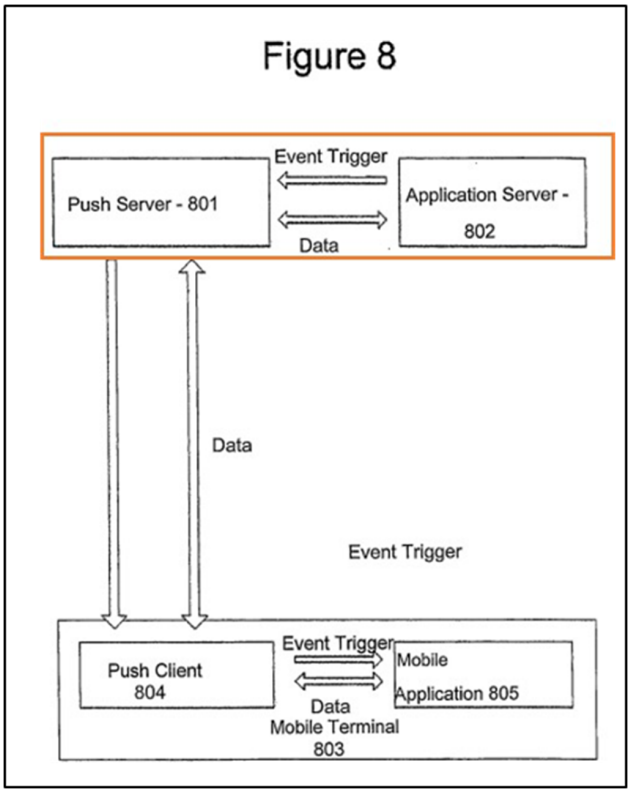
809. For the above reasons, Houghton-Munson renders obvious 1[b1].

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

810. Houghton discloses or renders obvious a network interface to receive messages (e.g., an application programming interface (API)) from a plurality of network elements (e.g., application server(s) 702/802).

811. In more detail, Houghton's Figure 8 (reproduced below) illustrates an example of a "COMMAND PUSH" procedure where the 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 in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705***." Ex-1007, 21-22.

812. Fig. 8 of Houghton is shown below (annotated):



Id., Fig. 8.

813. Thereafter, "a data connection between application server 802 and mobile application 805 is established with the aid of the persistent managed, tested and configured data connection in Figure 4, arrow C between push server 401/801 and push client 405/804" and "**[a]*n IP or other API (application programming interface) connection between application server 802 and push server 801 simplifies the work and cost of creating application server 802 by taking care of network details already established for the COMMAND PUSH procedure.*" *Id.*, 22-23.**

814. Houghton further discloses that, "[i]n some cases a separate server such as a telecommunication service server, gaming environment server, and original equipment manufacturer device or business information system may notify the server directly of events or messages to be sent using ***an external application programming interface (API)***. Such a device ***connected by external API*** may also receive information including the results and responses of messages which the user has replied to by terminal software or web page interaction." *Id.*, 14; *see id.*, 21 ("The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 ***and direct such to the push server 401, possibly for redirection through push server API*** to other servers, mobile terminals, push clients and computing devices.").

815. Moreover, the application server communicates over a network using the API or IP connection with push server 401. *See id.*, 23 ("***An IP or other API (application programming interface) connection between application server 802 and push server 801*** simplifies the work and cost of creating application server 802 ***by taking care of network*** details already established for the COMMAND PUSH procedure").

816. Indeed, before the Critical Date, it was well known for application servers and push servers to be distributed across a network, and for such servers to communicate over a network using an interface such as an API.

817. Additionally, a POSITA would have understood or found it obvious that the push environment contemplated by Houghton would include multiple different application servers, each separately communicating with the push server to exchange data/messages with one or more mobile terminals that are connected to push server 401.

818. This is further corroborated by Houghton's following disclosure: "The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 and direct such to the push server 401, possibly ***for redirection through push server API to other servers***, mobile terminals, push clients and ***computing devices***." *Id.*, 21.

819. Houghton also discloses that the message includes "a data packet" with "information specifying which mobile application 406 from" multiple "applications" is the message recipient (in other words, that the recipient application is identified in the message). *Id*., 21 (disclosing that the push client can accept "data D"); cl. 15 ("the push server sends to the push client…[an] application code, application code updates or application data updates"). Before January 28, 2009, it was well-known for such information specifying/identifying a particular application to be an application identifier. Ex-1006, [0011] (describing a "push service system" in which "push information" includes, *inter alia*, "push data" and "an associated application ID specifying the application for invoking the push data") [0013],

[0022], [0023] (push message includes application's "app_ID"), claim 1. I also describe in further detail in 1[e3] (which I incorporate here) how Houghton identifies applications in its messages.

820. In my opinion, a POSITA would have understood or found it obvious that Houghton's push server includes an interface to a network in the form of one or more APIs via which application-specific messages and data are received, from one or more network elements such as application servers. Moreover, a POSITA would have understood that application servers are network elements and hence messages received from such servers would constitute network element messages.

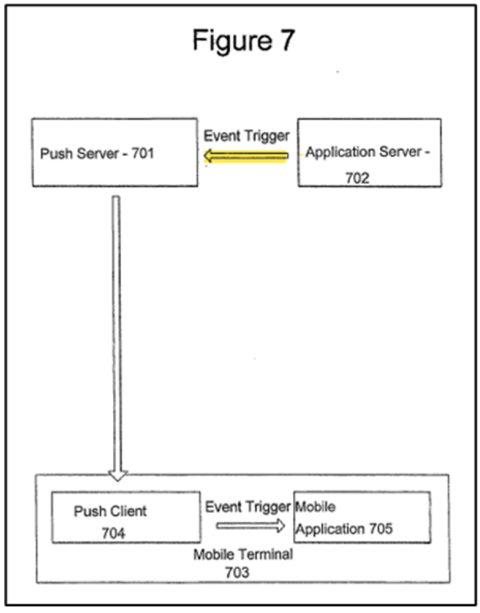
821. For the above reasons, Houghton-Munson renders obvious 1[c1].

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

822. Houghton discloses or renders obvious that network elements are authorized to send messages via the link interface to one or more software components on one or more of the wireless end-user devices. Houghton's push server (401/801) receives push messages from an application server (e.g., 802) and sends them to one of a plurality of mobile applications (authorized software components) through push client (405/804) on a mobile terminal (404/803). Ex-1007, 21-22.

823. In more detail, in Houghton, "the push server 701 is triggered by a trigger event as for example from an application server 702 to push an application command message to the push client 704 and thereby initiate a mobile terminal client trigger event in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705." Ex-1007, 21-22.

824. Upon occurrence of this trigger, "a data connection between application server 802 and mobile application 805 is established with the aid of the persistent managed, tested and configured data connection in Figure 4, arrow C between push server 401/801 and push client 405/804." *Id.*, 23.



Id., Fig. 7.

825. "The trigger mechanism may be between separate parts of a larger application or application suite, or may be an IP-triggered application launch and

data transfer as in the case of mobile application launched by contact to a specified IP port." *Id.*, 22. The "push server 901 is triggered by a trigger event of which the application server 902 is aware to send push message to the mobile terminal 905," which causes "pushing application launch commands, lazy application updates, and lazy application data updates through push clients 904 and 905 to mobile application 907 through local command and data path D (also Figure 5, arrow A)." *Id.*, 24.

826. From these disclosures, a POSITA would have understood or found it obvious that the received push messages from the application server request delivery of message content to particular mobile terminals, and in particular, to the applications resident on those mobile terminals.

827. Moreover, because push messages are routed to a particular application among multiple applications, and the destination application operates on the received message data (e.g., launch application, present content, etc.), in my opinion, a POSITA would have understood or found it obvious that applications are authorized to send data included in the command push/application command messages.

828. However, even if Houghton does not expressly disclose authorizing applications to send and receive push messages, a POSITA would have considered that obvious for the same reasons I describe above with respect to 1[b2] (which I incorporate here). A POSITA would have found it obvious to implement well-known

authorization/registration processes within Houghton to enable device and associated application registration with a push client and/or push server.

829. For the above reasons, Houghton-Munson renders obvious 1[c2].

vi. ***1[d1]-[d2]:*** *a message buffer system including a memory and logic, the memory to buffer content from the received network element messages for which delivery is requested to any of the wireless end-user devices;*

830. Houghton-Munson renders obvious a message buffer system including a memory and logic, where the memory buffers content from the received network element messages for which delivery is requested to a given one of the wireless end-user devices.

831. As I explain in the combination of Houghton and Munson (and incorporate here), in my opinion, a POSITA would have found it obvious in view of Munson to implement a message buffer in Houghton's push system to store received push messages on Houghton's push server. In more detail, and as explained with reference to 1[c1]-1[c2] (which I incorporate here), Houghton-Munson render obvious that the push server receives network element messages in the form of push messages from application servers, where delivery of such messages is requested to one or more mobile terminals/wireless end-user devices.

832. The push server further includes a message buffer system that includes logic (as described with reference to 1[d3] and 1[d4] below) (and which I incorporate

here) and, in view of Munson, would have also included memory that buffers content from the received push messages (for the reasons stated in the Combination of Houghton and Munson, above, which I incorporate here).

833. Indeed, Houghton discloses that push messaging services, like "SMS (Short Message Service)" and "MMS (Multi-media Messaging Service)" "store and forward" push messages. Ex-1007, 3, Figs. 3-4. Houghton further discloses that servers (such as push servers) "control and deliver messages and store messages when the mobile terminal is unavailable" and then later deliver those stored messages when "the terminal becomes available." Ex-1007, 7.

834. Given (1) messages may be received by the push server when the client device is unavailable and (2) Houghton's express disclosure of storing such messages until the user becomes available, a POSITA would have understood or found it obvious that the push server would include a buffer/memory for storing received messages.

835. Indeed, in the similar context of push communications, it was well-known, as taught in Lee, to store push messages "when the mobile terminal is not connected to the mobile internet service network" and "re-transmitting the stored push information when the mobile terminal is connected to the mobile internet service network." *See* Ex-1006, cl. 14, Fig. 1 (showing a push information storage device 112 for storing push messages). Lee also contemplates that a scheduler 114

can "schedule and transmit push information push_info or retransmit the push information push_info." *See id.*, [0034], Fig. 34.

836. Consistent with these well-known teachings of storing push messages in push server's memory/storage, Munson, which also enables push messages, discloses a "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 until all contents in a series arrive." Ex-1017, [0036]-[0037], Fig. 4. Munson's Figure 3 illustrates a structure for queuing messages in a Content Push Service 220. *Id.*, [0035]-[0036], Fig. 3.

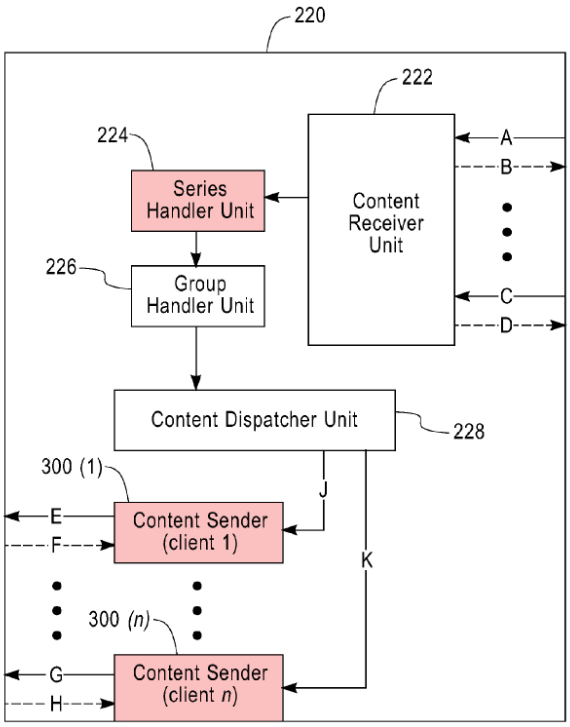


FIG. 3

Id., Fig. 3.

837. Munson additionally discloses a plurality of "Content Sender[s] 300" that each contain a "Queuer 320" and "Push Queues 330." Ex-1017, [0036], [0038], Figs. 3, 5. Munson discloses that when pushing messages, Content Senders 300 check "urgency level of contents … customer level … [and] dequeue policy." *Id*. Moreover, each "client device" (e.g., the Houghton-Munson mobile terminal) is assigned a Content Sender 300. *Id.* Munson's Figure 5 illustrates a structure for queuing messages in a Content Sender 300. Ex-1007, Fig. 5.

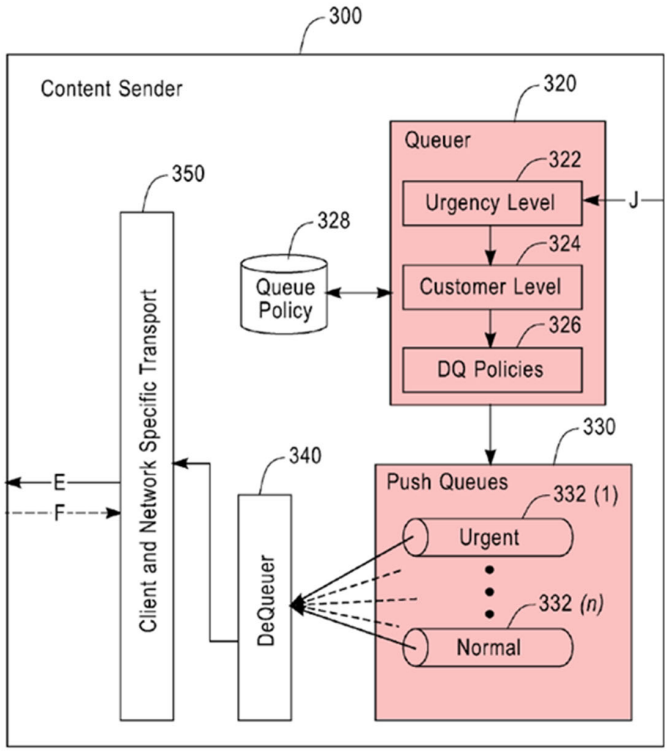


FIG. 5

Ex-1017, Fig. 5.

838. In view of the above-described teachings in Munson and Houghton, and for the reasons stated in the Combination of Houghton and Munson, above, which I incorporate here, a POSITA would have found it obvious to implement Munson's above-described message buffering functionality and associated components as part of Houghton's push server to enable storage and later forwarding of messages to one or more mobile terminals/push client, e.g., upon occurrence of certain trigger events. Thus, in the Houghton-Munson push server, the received network elements messages would be stored in the buffer and associated components, as described in Munson and as implemented in Houghton's push server.

839. Additionally, as described below (1[d3]-1[d4], which I have incorporated here), Houghton's push server includes a message buffer system with logic that facilitates delivery of messages upon occurrence of one or more message delivery triggers.

840. For the above reasons, Houghton-Munson renders obvious 1[d1]-1[d2].

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

841. As explained below, Houghton-Munson renders obvious the message buffer system including logic that determines when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred, where the receipt of a message is not a message delivery trigger and one of the message delivery triggers is an occurrence of an asynchronous event with time-critical messaging needs (e.g., a user request for a received message or the occurrence of a transaction).

842. In more detail, Houghton contemplates the occurrence of trigger events to trigger sending push messages to the client devices. For example, Houghton discloses that the "***push message from the push server 401 may be triggered by any trigger event, local or remote, defined at the server. The trigger may, among others, include an alarm, notification, or measurement result received to the push server 401 from another device or system***." Ex-1007, 21; *see id.*, 14 (describing various trigger events), 21-22, 25-28 (describing various trigger events).

843. Houghton therefore discloses that the delivery of push messages can be triggered upon the occurrence of certain trigger events and that the trigger event can be "local or remote" and may include "an alarm, notification, or measurement result received" by the push server 401 from another device or system. *See id.*, 21. Thus, a POSITA would have understood or found it obvious that Houghton's system would implement message delivery triggers for triggering delivery of messages to the wireless end-user devices.

844. Moreover, the triggers noted above—alarm, notification, or measurement result—are message delivery triggers and do not constitute a trigger that is based on receipt of the message by the push server.

845. Additionally, Houghton discloses or renders obvious that the message delivery triggers include an occurrence of "an asynchronous event with time-critical messaging needs," consistent with how this claim phrase is described in the '320 patent's specification. Specifically, and as explained with reference to this limitation in Ground 1A above (which I incorporate here), the '320 specification explains that an asynchronous event with time-critical messaging needs encompasses a "user request" for the delivery of the message or the occurrence of a transaction. Ex-1001 ('320 Patent), 38:50-64.

846. Consistent with this description, Houghton renders obvious a message delivery trigger that is an occurrence of an asynchronous event with time-critical

messaging needs. Specifically, and as noted in the above-cited disclosures, the trigger event can be an alarm or notification received from another device.

847. Houghton further contemplates that message delivery can be adjusted based on whether the message to be delivered is time-critical or "non-time-critical." *See, e.g.*, Ex-1007, 20-21 (contemplating "time critical and the fastest available"), 23-24 ("[t]he push server 501 on becoming aware of such a situation may inform (arrow D) the application server 502 of the availability of a more desirable network connection (C plus A) to the client and on behalf of the application server 502 or independently decide to push additional commands and data to the wireless terminal using this connection. Further, the push client in the broadband access point has the additional capability to route such information to the wireless terminal. ***The result benefits customers using the wireless terminal in that data is managed and updated automatically for their mobile applications in the most cost efficient manner for non-time-critical capabilities such as lazy application code update and lazy application data update***. While such alternative routing may also be achieved through normal IP routing and path cost without the presence in the broadband access point 509 of push-aware client software, ***the push server 501 and application server 502 would in this known-in-the-art implementation not be aware of when the low cost alternative route was available and thus able to at such time trigger the push of lazy application code and data installation and updates***."). A POSITA

would have therefore found it obvious to modify message delivery based on time criticality of the message and therefore, adjust message delivery and the associated trigger in this context.

848. Similar to Houghton, Munson discloses that its push system provides "***asynchronous content push[es]*** (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" (Ex-1017, [0040], [0044]), thus rendering it obvious that a message delivery trigger could be "an asynchronous event with time-critical needs" as claimed. In my opinion, therefore, even if Houghton's triggers do not disclose this limitation, the combination of Houghton and Munson does because Houghton contemplates configuring the system for "any trigger event," which would have included those described in Munson as additional example triggers in a similar networking system. Ex-1007, 21.

849. Houghton's push client can also send a message based on a user request to the push server, requesting delivery of a push message. Specifically, "IP push command messages" may be triggered when the "push client makes" the push server aware of certain "client-side events," which in turn trigger the push server to deliver push messages intended for particular mobile application(s) 406 residing on the mobile terminal. Ex-1007, 14, 21. For example, the "client-side events" including, among others, "***creation of photographs video or other media***," "***video game***

actions or events," and ***"messaging actions***" are triggers that "***push client makes the server aware of***" so that the server pushes a message to the mobile terminal and a particular application residing therein. Ex-1007, 14. Houghton further discloses that the trigger events include "***remote application user*** or application server ***event***, ***remote user action***." *Id.*, cl. 22.

850. Given the above disclosures, in my opinion, a POSITA would have understood or found it obvious that these client-side events are user requested events or transactions that occur, e.g., as a result of transactions and events occurring on a mobile terminal, including user actions taken with respect to creating media, gaming events, or messaging.

851. A POSITA would have thus understood or found it obvious that such user actions or transactions trigger the delivery of messages and therefore, constitute message delivery triggers that constitute an asynchronous event with time-critical messaging needs, as that phrase would be understood in the context of the '192 Patent.

852. Because the push server waits for these triggers to occur before it delivers the message to push clients, a POSITA would have understood or found it obvious that the push server includes logic that determines when one (or more) of these message delivery triggers for the particular terminal/end-user device has occurred and if so, delivering the messages. In my opinion, a POSITA would have

therefore found it obvious, in view of Houghton or Houghton-Munson, to adjust message delivery based on the message's time criticality and therefore, adjust message delivery and the associated trigger on an asynchronous basis.

853. For the above reasons, Houghton-Munson renders obvious 1[d3].

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

854. Houghton-Munson render obvious that, upon determining that one of the message delivery triggers has occurred (see above analysis for 1[d3] which I incorporate here), the logic further to supply one or more messages comprising the buffered content (see above analysis for 1[d1]-1[d2] relating to storage of received push messages) 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.

855. As I described in 1[a]-1[b1] (which I incorporate here), Houghton's push server would have included a "link interface" (e.g., including a stack of transport protocols, *see* Ground 1A (noting similarities between claimed "link interface" and "transport services stack") that facilitates a secure message link

between the push server and the push client (on the mobile terminals). Houghton further discloses, in view of Munson and as described with reference to 1[d1]-1[d2] (which I incorporate here), to store the received push messages in push server's memory. In my opinion, as I explained in the 1[d1]-1[d2] , a POSITA would have appreciated Munson's content senders operate similarly to Houghton's routing to transmit messages from the buffer to the client device. Ex-1017, [0036]-[0037].

856. In the combination, the push server would include logic, and the stored push messages (and the associated data/content) would be delivered to the push client upon the occurrence of one or more message delivery triggers, as described with reference to 1[d3] (which I incorporate here).

857. Specifically, Houghton discloses "intelligence" that "route[s] messages received on behalf of the wireless terminal to and from the server." Ex-1007, 14. Such message delivery would happen using the secure message link already established/maintained between the push server and the push client of the mobile terminal. *Id.*, cl. 1 ("the push server sends, in response to predetermined trigger events, at any time push application installation, updates, commands or data messages to be directed to a push application or dynamically loaded module or user interface feature from among a plurality of such applications, modules and features by means of said client-initiated or client-opened and client-maintained permanent connection").

858. For the above reasons, Houghton-Munson renders obvious 1[d4].

ix. ***1[e1]:*** *the device link agents on the respective wireless end-user devices, each of the device link agents configured to;*

859. Houghton discloses or renders obvious a device link agent on a wireless device.

860. As I explained in 1[b2] (which I incorporate here), Houghton discloses a plurality of mobile terminals, each with a push client (Ex-1007, 21 (noting connections "to other…mobile terminals"), 23 ("users of mobile terminals")), which are configured consistent with 1[e2]-[e4] (which I incorporate here).

861. For the above reasons, Houghton-Munson renders obvious 1[e1].

x. ***1[e2]:*** *maintain the respective secure Internet data message link over a wireless network to the link interface;*

862. Houghton discloses or renders obvious that the device link agent (e.g., push client) is configured to maintain a secure link over a wireless network to the link interface.

863. As discussed in 1[b1] and 1[c1] (which I incorporate here), Houghton discloses that the push client "initiates and maintains…[a] data connection…to the push server using Internet technologies." Ex-1007, 11, cl. 1 ("the push client initiates and maintains over an extended period of time a packet-based data connection through at least one communication security or inter-network connection device of

the serving wireless system to the push server using Internet technologies"). In Houghton, the push client connects to the push server "over a data network" (e.g., "a wireless network" for "transmitting Internet Protocol (IP) packets"). *Id.*, 17, Fig. 4.

864. For the above reasons, Houghton-Munson renders obvious 1[e2].

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

865. Houghton-Munson renders obvious that the device link agent (e.g., push client) is configured to receive secure messages from the network server system (e.g., push server) over a secure link, including messages collected from multiple network elements and messages corresponding to multiple software components authorized to receive messages via the device link agent, where a subset of messages include both a unique identifier for the software agents and data to be consumed by that software component provided by a respective network element.

866. As I discussed in 1[b1] through 1[d4] (which I incorporate here), Houghton discloses that the push client receives messages from the push server using a variety of protocols, including "secure protocol[s]" such as "HTTPS, IP-Sec, secure IP6 or a proprietary security protocol". Ex-1007, 18-20. As a I explained in 1[b1], because messages may be transmitted using "secure protocol[s]," in my opinion, a POSITA would have understood that the messages constitute "secure Internet data messages" as claimed. Moreover, Houghton discloses push clients may receive messages from a variety of other servers and push clients and (registered) applications, as I described in 1[b1] through 1[c2] (and also incorporate here).

867. Houghton also explains that the message sent from an application server to an application on the user's terminal through push server 401/801 and push client 405/805 includes "a data packet" along with "[a]dditional binary or text information ***specifying which mobile application 406 from a plurality of such applications may optionally be*** included" as well as "***binary or text information to be passed to the default or specified mobile application***" **and** ***"additional binary or text information specifying how upon receiving such a message the client 405*** will notify and optionally ask the user if the action should be performed may optionally be included." *Id.*, 21; *see id.*, claim 1 ("***pushing commands and data to a wireless terminal*** from a push server" to a push client"), 8 ("When a remote program such as a push server 202 sitting inside the operator's firewall 201 and with permission to

contact ports on the mobile terminal 203 sends an IP packet to the terminal (A), ***the IP terminal software 204 routes the packet with [its] commands or application-specific data to the mobile application 205 (B), launching the application if it is not already running***."), 21 ("Upon receiving such a message***, appropriate automated actions and user interface media capabilities of the mobile terminal as specified in the command C from the server 401 are executed by the push client 405. The most common case is playing a sound, asking the user if the mobile application should be launched, and if affirmative passing the command details (arrow D) to the specified mobile application,*** module or additional feature within the same application suite, launching the application, module or feature if needed. Such action may include optionally proceed without audio, with network downloaded audio, or without first requesting confirmation from the user as specified by user preferences and command details."). There could also be a "data packet containing no information." *Id.*, 21.

868. In one particular example, Houghton discloses a "trigger event" or "application command message" issued by the application server, which triggers the push server to transmit an application message that 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" in the message. Houghton, 21-22.

869. By stating that the message "specif[ies] which mobile application 406 from a plurality of such application" to which the message is directed, (*id.*, 21) and that the message is directed to a particular "mobile application 705 from a plurality of such applications" (*id*., 22), in my opinion, a POSITA would have understood that Houghton's messages include a unique identifier for the recipient application.

870. But even if Houghton did not expressly include an application identifier, in my opinion, including one would have been obvious. Before the Critical Date, it was well-known for such information specifying/identifying a particular application to be in the form an application identifier. Ex-1006, [0022]. For example, in the relevant context of push communications and associated environments, Lee discloses that the push message from a push server includes an "app_ID" that specifies a target application on the mobile terminal that is the destination for the message. *Id.*, [0013], [0022]. Similarly, TS-23.140 also contemplates that each destination application to which the push/MMS message is routed is identified using an application identifier in the message. Ex-1004, 54-56.

871. Given that Houghton routes messages to an application based on the "information specifying … [that] application" and given the common and well-known use of application identifiers to facilitate such identification, a POSITA would have found it obvious to implement Houghton's teaching of the "information specifying" the mobile application as the application identifier for that application.

The messages received by the push server from the application server would include an application identifier because the application server is better equipped, relative to the push server, to provide the application identifier since the application server originates the message and knows the intended application.

872. Implementing Houghton's system in this manner therefore would have amounted to an implementation of 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, as suggested by Houghton and as was well-known in the art before the Critical Date).

873. Taken together, in my opinion, a POSITA would have understood that messages directed to specific applications sent over a secure link are "a subset of secure Internet data messages" as claimed, which include both a "unique" destination application identifier to identify the intended application and application data intended for that application supplied by the originating network element.

874. For the above reasons, Houghton-Munson renders obvious claim 1[e3].

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

875. Houghton-Munson renders obvious that, for software components that are authorized to access messages received via the device link agent (e.g., push client), the device link agent causes messages with a unique identifier corresponding to an application to be securely delivered to that application.

876. As I discussed in 1[b1], 1[c2], and 1[e3] (and which I incorporate here), Houghton discloses that the push server transmits application-specific messages—which contain application identifiers—to push clients on mobile terminals using "secure protocol[s]," and the push clients route those messages to registered applications also on the mobile terminals. Because application-specific messages are transmitted to the push client using a secure link and routed to a registered application, in my opinion, a POSITA would have understood they are "securely delivered to a software process corresponding to the given software component" as claimed. Houghton also discloses that the push client can pass "command details (arrow D) to the specified mobile application, module, or additional feature within the same application suite" for processing by the application. Ex-1007, 21, Fig. 4, cl. 1.

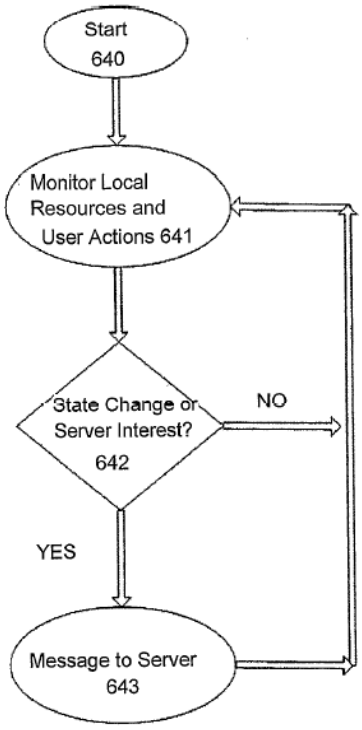
877. For the above reasons, Houghton-Munson renders obvious claim 1[e4].

b. ***Claim 2:*** *The networked system of claim 1, the message buffer system logic to determine, for each of the wireless end-user devices, when one of a plurality of message delivery triggers has occurred, at least one of the triggers for each given device specific to one or more states of that given device.*

878. Houghton discloses that one of the message delivery triggers is the state of the given device.

879. As I discussed in 1[d1]-1[d4] (and which I incorporate here), Houghton discloses the logic evaluates a set of trigger conditions, one of which "a state change (a trigger event)." *Id.*, 26. According to Houghton, "if a state change…is detected…, the push client 405 may send a message to the server 401." *Id.* at 26, 14 (explaining that triggers could be "changes in [the] internal state of electronic equipment"), cl. 22 (explaining that trigger events include "one or more of:…computing equipment change of state"), Fig. 6D.

Figure 6D



Ex-1007, Fig. 6D.

880. For the above reasons, Houghton-Munson renders obvious claim 2.

c. ***Claim 3****: The networked system of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

881. Houghton-Munson renders obvious that one of the message delivery triggers is the expiration of a periodic timer.

882. Houghton contemplates multiple message delivery triggers (Ex-1007, 14, 21) and additionally discloses a periodic message (which would obviously be triggered sent at "the expiration of a periodic timer " as claimed) that ensures that devices in the communication path (i.e., push client at mobile terminal and push

server) "do not time expire the connection," suggesting that the periodic messages in Houghton may recur regularly. *Id.*, 26, 19 ("Use of a periodic message from the client to the server or server to the client to the effect that client socket, server socket, intermediate Network Address Translation (NAT) devices, intermediate firewalls, and intermediate gateway devices do not time expire the connection."), 11 ("In an embodiment of the invention, the mobile terminal maintains contact with the server by periodically testing the connection with a suitable keepalive message, a 'ping', which resets timers in intervening operator stateful firewalls and/or NAT (network address translation) firewalls. The mobile terminal may also automatically re-establish the connection when the connection gets broken and adjust its connection settings based on past experience with failure to connect and such broken connections in a given mobile network.").

883. Houghton explains that the "periodic message timings" can be "measured in seconds, minutes or hours" and can be "longer or shorter…to match the combined needs of all such network devices and protocols." *Id.*, 19-20, 26. Additionally, the timing between periodic messages can be "adjusted based on the success or failure of earlier messages," also suggesting that these messages could recur at regular intervals. *Id.*, 19.

884. Munson also discloses that "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 to take advantage of lower rates." Ex-1017, [0044]. As I explained previously in 1[d3] (and incorporate here), a POSITA would have found it obvious to configure Houghton to implement Munson's triggers.

885. Given Houghton's disclosures of delivery triggers and Munson's use of periodic timers for message delivery, in my opinion, a POSITA would have been motivated and would have had a reasonable expectation of success in implementing Munson's teaching of using periodic timers (and expiration of the same) as a message delivery trigger, within Houghton's system that already contemplates using periodic timers. This combination would have been sufficient to cause message delivery. A POSITA would have been motivated to use periodic timers as a message delivery trigger, as contemplated by Munson, to enable more efficient use of network resources and avoid repeated message requests that can impact performance of network communications. A POSITA would have understood or found it obvious that "schedule[d] pushes during off-peak hours for maintenance would be triggered by the expiration of a regularly-repeating timer.

886. For the above reasons, Houghton-Munson renders obvious claim 3.

d. ***Claim 4****: The networked system of claim 3, wherein the period of the timer is fractionally shorter than a maximum data message interval beyond which the Internet data message link to the given device is taken down.*

887. Houghton-Munson renders obvious that the period of the timer is fractionally shorter than a maximum data message interval after which the secure message link is taken down.

888. Specifically, Houghton discloses "the push client 405 may send a periodic message…from the client 405 to the server 401 to effect that…[the network devices] do not time expire the connection. The period of time…between such periodic messages is set to match the combined needs of all such network devices and network protocols". Ex-1007, 26. Houghton discloses that the periodic messages are sent to ensure that devices in the communication path (i.e., push client at mobile terminal and push server) "do not time expire the connection," suggesting that the periodic messages in Houghton could recur regularly. *Id.*, 26, 19-20. A POSITA would have understood that this period sets a "maximum data message interval beyond which the secure message link is taken down" as claimed because the purpose of the message is to *prevent* expiration of the connection; failing to receive a period message will therefore terminate the connection.

889. In my opinion, a POSITA would have understood that the period to send a regularly recurring timer for triggering message delivery (which I again describe in claim 3 and incorporate here) would be some degree shorter than the

recurring time for Houghton-Munson's periodic messages (e.g., the claimed "maximum data message interval beyond which the secure message link is taken down"). If the periodic polling/message delivery was set to a longer period than the keep-alive timer period, the connection may be terminated before the periodic polling/delivery message could be sent, thus requiring a new connection to be established. And because the '320 Patent does not specify what constitutes "fractionally shorter," in my opinion, any degree shorter can be considered "fractionally shorter," as claimed, since it would be some "fraction" of the duration of the keep-alive timers.

890. In my opinion, because Houghton discloses both a periodic timer *and* periodic (connection) messages, a POSITA would have understood that Houghton discloses a periodic message set for a period of time that is shorter than a connection timeout after which the client and server are disconnected.

e. ***Claim 5-6: [5]*** *"The networked system of claim 1, wherein one of the message delivery triggers is the receipt of a transmission on the respective secure Internet data message link from the device link agent of the given one of the wireless end-user devices, or a response generated to a transmission received from that device link agent;* ***[6]*** *The networked system of claim 1, wherein one of the message delivery triggers is a heartbeat message generated by the given device link agent, or a request received from the given device link agent.*

891. Houghton discloses that one of the message delivery triggers is the receipt of a transmission over a secure link from the device link agent.

892. As described above, Houghton provides multiple examples of message delivery triggers that include "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," where the transmission is a request received from the given device link agent."

893. For example, Houghton describes that "client-side events" can result in "pushing application commands." Ex-1007, 14, 21, 16 ("Figure 4 illustrates an exemplary implementation of the system ***for pushing application commands to a mobile terminal based on*** server side and ***client side events*** …."), Fig. 4. Trigger events include "messaging actions" or a "notification." Ex-1007, 14, 21. Houghton's "client" notifies the "server" of a "terminal event" that triggers a "return message." *Id.*, 20-21. Houghton further discloses that "[i]f 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. Specifically, Houghton also describes that the "client" can notify the "server" of a "terminal event" that triggers a "return message." Ex-1007, 20-21.

894. Additionally, Houghton contemplates that a "connectionless protocol, such as UDP/IP," may be used "when the ***client 405 notifies the server 401*** of a[] terminal event, user interface event or application event" and the ***push server "in return 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." Ex-1007, 20-21.

895. In the context of established IP connections, the "push client 405 may monitor the local resources for a state change (a trigger event) in step 641. If a state change of other event is detected in step 642, the push client 405 may send a message to the server 401 (step 643)." Ex-1007, 26.

896. Moreover, as described above with reference to claim 3, an example message trigger includes periodic polling to ensure that devices in the communication path (push client at mobile terminal and push server) "do not time expire the connection," which a POSITA would have recognized or found it obvious as being akin to a heartbeat message generated by the given device link agent. Ex-1007, 19, 21, 26; Ex-1001 ('320 Patent), 38:50-63, 70:6-71:16.

897. For the above reasons, Houghton-Munson renders obvious claims 5-6.

f. ***Claim 7:*** *The networked system of claim 1, wherein one of the message delivery triggers is the receipt of a particular message from one of the network elements.*

898. Houghton-Munson renders obvious that one of the message delivery triggers is the receipt of a particular network element message from one of the network elements (e.g., a message from a particular application server). *See, e.g.*, Ex-1007, 21-22 ("Figure 7 illustrates an example of [a] 'COMMAND PUSH' procedure ***wherein the push server 701 is triggered by a trigger event as for***

example from an application server 702 to push an application command message to the push client 704 and thereby initiate a mobile terminal client trigger event in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705."), Fig. 7.

899. For the above reasons, Houghton-Munson renders obvious claim 7.

g. ***Claim 10:*** *The networked system of claim 1, wherein a second subset of the secure Internet data messages contain a user message from a network element, the user message intended for display on a user interface of a given one of the wireless end-user devices.*

900. Houghton discloses a subset of messages are intended to be displayed on a user interface of the user's wireless device.

901. As I explained above regarding 1[e3] (which I incorporate here), Houghton discloses secure Internet data messages. At least some messages may include "a message stimulating…[the] display or shift to [the] foreground of a user interface feature of the terminal such as an application" or a "display notification." Ex-1007, 11-12, cl. 18 (noting messages can include an "event push message triggering the wireless terminal…to display notification information such as text, rich text, graphics and rich media from which the user may select or decline to launch the application"), Fig. 6C. In my opinion, a POSITA would have understood that messages displayed to the end user would be a "user message" as claimed.

902. For the above reasons, Houghton-Munson renders obvious claim 10.

h. ***Claim 11:*** *The networked system of claim 1, wherein one of the software components on a given one of the wireless end-user devices is a policy control agent, and the data to be consumed by the policy control agent comprises service settings and/or configuration information for the given device.*

903. Houghton discloses or renders obvious that one software component on a wireless device is a policy control agent that consumes service setting or configuration information for that device.

904. Houghton discloses that the push client "is also capable of receiving and storing variables which configure behavior of the client," and "may thus specify the network connection preferences, notification sounds, and other behavior of the client." Ex-1007, 22. In my opinion, a POSITA would have understood, therefore, that Houghton discloses, or at the very least suggests, some software component on the client that receives and then implements control information (which would serve as a "policy control agent" as claimed) including connection preferences and other configurations from the server.

905. For the above reasons, Houghton-Munson renders obvious claim 11.

i. Claim 15

i. ***15[a]-15[b]:*** *wherein the device link agent on a given one of the devices is further configured to receive, from one or more of the software components on that device, upload messages, each of the upload messages identifying a corresponding one of the network elements to which the device respective software component has requested delivery*

906. Houghton-Munson renders obvious that the device link agents (*push clients*) receive upload messages from one or more of the software components and that identify which network element has requested delivery.

907. In more detail, Houghton discloses that push client 405 "at any time accept[s] ***data D from each of a plurality of mobile applications 406 and direct[s] such to the push server 401***, possibly for redirection through push server API to other servers, mobile terminals, push clients and computing devices." Ex-1007, 21; *see id.*, 26 ("***the client 405 may also send a message to the server 401 (step 633),*** such as a command execution acknowledgement or result"). The "messages received on behalf of the wireless terminal" may be routed "to and from the server." Ex-1007, 14. The message's "data packet" could include "information specifying which mobile application 406 from a plurality of such applications" to which it will be directed to. *Id.*, 21.

908. Moreover, given that Houghton's push client uses a stack of secure protocols (as described with reference to 1[b1] above and which I incorporate here),

and may encrypt those connections, a POSITA would have understood or found it obvious that those protocol-based connections would receive messages (i.e., upload messages) on the push client (i.e., respective device link agents) from some of the applications 805 (i.e., software components on the device) for forwarding to the push server

909. Additionally, as described above, a POSITA would have understood or found it obvious that each such message would identify, e.g., the particular application server (network element) to which the application issuing the message (software component) requested delivery. *See also* Ex-1006, [0011] (describing a "push service system" in which "push information" includes, inter alia, "receiving mobile terminal information specifying the mobile terminal receiving the push data"—i.e., the identifier for the device to which the push message is directed), [0022], claim 1.

ii. ***15[c]:*** *the device link agent on that device transmitting the upload messages to the network message server over the respective secure Internet data message link, for delivery by the network message server to the respective identified network elements.*

910. First, I note that claim 15 uses the term "network message server" that does not appear elsewhere in the claims. However, I have assumed that the "network message server" refers to the "network server system" that appears in claim 1. With that understanding, Houghton discloses or renders obvious that the device link agent

on a device transmits upload messages to the network message server over the secure link for delivery by the network message server to the identified network element(s).

911. As described with reference to 15[a]-15[b] and claim 1 (which I incorporate here), Houghton discloses that the push client (i.e., device link agent) transmits the messages to the push server (i.e., the network message system) across a link using a stack of "secure protocol[s]" (i.e., secure Internet data message link) for delivery by the push server to the identified recipients (i.e., respective identified network elements).

912. For the above reasons, Houghton-Munson renders obvious 15[a]-15[c].

j. ***Claim 18:*** *The networked system of claim 1, wherein at least one of the secure Internet data messages comprises multiple identifier/data pairs.*

913. Houghton-Munson renders obvious that at least one of the one or more messages comprises multiple identifier/data pairs.

914. As I described in 1[b1] and 1[b2] (which I incorporate here), Houghton's push server includes a stack of secure protocols for API or IP-based connections between the push server and the application server to establish a connection between mobile terminals and the push server.

915. As I also described in 1[b2] and 1[c2] (which I incorporate here), push messages would be received by the stack of secure protocols (link interface) implemented as part of Houghton's push server from different network elements

(e.g., application server or other push clients resident on a respective mobile terminal), and each such message intended for an application executing on one of the mobile terminals would include data and corresponding identification of the intended software application. *See also* Ex-1006, [0011] (describing a "push service system" in which "push information" includes, inter alia, "push data" and "an associated application ID specifying the application for invoking the push data") [0013], [0022], [0023] (push message includes application's "app_ID"), claim 1.

916. Based on these teachings, a POSITA would have recognized or found it obvious that messages received from the push server would have been directed to one or more applications from among the "plurality" of applications ("***multiple identifier/data pairs***") on a user's mobile terminal because Houghton discloses various "actions" that occur across applications when receiving a push message. Ex-1007, 21. Specifically, the message's "data packet" may include "information specifying which mobile application 406 from a plurality of such applications" it will be directed to. *Id.*, 21. This is corroborated by Houghton, which discloses that, in some cases, the "packaging of mobile applications involves combining multiple applications delivered in a single bundle" (e.g., a "bundle" or "suite" of applications would receive messages packaged together – a message containing "***multiple identifier/data pairs***"). *Id.*, 12.

917. This therefore demonstrates that the receipt of multiple identifier/data pairs in the one or more messages for delivery by the transport services stack of the push server.

918. For the above reasons, Houghton-Munson renders obvious claim 6.

I. Ground 2B: Claim 1 Is Alternatively Rendered Obvious by Houghton, Munson, and TS-23.140

919. If Houghton-Munson does not disclose or render obvious claim 1—and specifically that applications must register/be authorized or that the received message includes an application identifier—in my opinion, it would have further been obvious to implement these features as taught by TS-23.140.

920. TS-23.140 teaches that applications must register before being granted access to the MMS network, during which "the application provisions an MMS User Agent or an MMS VAS Application with its application identification value and may negotiate with the MMS User Agent or MMS VAS Application the details (amount and format) of information to be exchanged between the two entities." Ex-1004, 54-55. In other words, the registration process is intended to ensure the messages the application intends to send using the MMS network are compatible with the network, and that the network will be able to transmit/receive application-specific messages as intended. TS-23.140 also discloses that application-specific messages sent by an MMS User Agent or an MMS VAS Application includes an "application identifier

of the destination application." Ex-1004, 55. The application identifier is used to direct application-specific messages to the intended target. *Id.*

921. In my opinion, a POSITA would have first been motivated to incorporate an application-registration process like TS-23.140's into Houghton-Munson's system to ensure each application is configured to transmit messages compatible with the network and ensure successful message deliveries. Ex-1004, 54. To implement this feature, for instance, an application would negotiate with a push client to establish: (1) the application's identity to the network; (2) the formats, procedures, and protocols to be used when transmitting incoming/outgoing messages to/from the application through the push client. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification because it merely involves well-known, negotiated signaling between the registering application and existing messaging-network element (such as Houghton-Munson's push client). In fact, TS-23.140 leaves the specific implementation of the registration process to those in the art to customize, which is an acknowledgement that those in the art knew how to implement specific registration processes.

922. After an application has registered and become authorized to access the network, in my opinion, a POSITA would have been further motivated to include application identifiers in application messages sent through Houghton-Munson's system, like TS-23.140 does, to ensure those messages are directed to the intended

application, or to verify a message received by, e.g., an application server was issued by a compatible application. To analogize, if a user device is thought of as an apartment building, and each application on the device is a different apartment, it is like putting the apartment number on a mailed letter; you would not simply send letters with the building number hoping it reaches the right apartment in the building. Nor would some company of authority (akin to an application server) accept and execute based on the contents of mail (like application-message contents) from senders who fail to identify themselves. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification as well because the combination merely involves incorporating an express "application identifier"—typically a small text string—into the message. *See* Ex-1004, 55 ("The format of the application identifiers' value shall be text string.").

923. Moreover, if a combination with TS-23.140 is necessary, in my opinion, the combination would also teach all other Challenged Claims set forth in Grounds 2A, and 2C-G for the reasons states therein, in combination with TS-23.140. The remaining combinations involve discrete teachings separate from an application identifier, so any further combinations would, in my opinion, not impact the inclusion of an application identifier like TS-23.140 in Houghton-Munson.

J. Ground 2C: Claims 3 and 4 Are Rendered Obvious by Houghton, Munson, and Adamczyk

1. ***Claim 3:*** *The networked system of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

924. Like Houghton-Munson, Adamczyk discloses a message-transmission system. *See, e.g.*, Ex-1021, Abstract, [0007]-[0009]. However, Adamczyk explains that 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 the preferences of the user. *Id.*, [0010], [0022], cl. 4.

925. Although Houghton-Munson discloses a series of triggers, as I explained in 1[d3] (and which I incorporate here), in my opinion, a POSITA would have appreciated that Houghton-Munson's triggers could be supplemented with the additional types of triggers like those expressly disclosed by Adamczyk. Specifically, a POSITA would have been motivated to provide users the option to schedule the delivery of messages—including on a "periodic schedule," as taught by Adamczyk, sufficient to control when messages would be pushed to their devices (one of the message delivery triggers is the expiration of a periodic timer).

926. In my opinion, a POSITA would have had a reasonable expectation of success in combining these two concepts because the Houghton-Munson combination already taught timers to control the delivery of messages and periodic messaging. Indeed, to configure the push server to push messages at periodic intervals, the combination would merely require configuring the push server to hold

received messages in the buffer until expiry of a recurring timer before pushing messages received during the time interval at once.

2. ***Claim 4:*** *The networked system of claim 3, wherein the period of the timer is fractionally shorter than a maximum data message interval beyond which the Internet data message link to the given device is taken down.*

927. Houghton-Munson-Adamczyk renders obvious that the period of the timer is fractionally shorter than a maximum data message interval after which the secure message link is taken down for the same reasons I have explained previously with respect to Houghton-Munson in claim 4 (which I incorporate here), and Houghton-Munson-Adamczyk in claim 3 (which I incorporate here). Specifically, POSITAs would have understood and found obvious that the period of the periodic timer trigger consistent with Adamcyzk would be fractionally shorter than the period for sending the "periodic message" that prevents connection expiration in Houghton-Munson for the same to prevent the connection from closing before triggering a periodic message delivery.

K. Ground 2D: Claims 6, 12, 13, and 16 Are Rendered Obvious by Houghton, Munson, and Shen

1. ***Claim 6:*** *The networked system of claim 1, wherein one of the message delivery triggers is a heartbeat message generated by the given device link agent, or a request received from the given device agent.*

928. In my opinion, to the extent Houghton does not disclose a heartbeat message, Houghton-Munson combined with Shen also renders obvious the use of a heartbeat message as a trigger.

929. As I explained in claim 3 (and incorporated here), Houghton discloses using a "periodic message" as a message-delivery trigger to keep a connection alive, and monitors for when the client (405) has become disconnected. Ex-1007, 19-20, 26-27. Although Houghton does not expressly describe this feature as a "heartbeat message," in my opinion, a POSITA would have understood that it is, as it performs identical functions as what those in the art called a "heartbeat message.

930. If, however, Houghton's "periodic message" cannot be considered a heartbeat, in my opinion, a POSITA would have been motivated to implement Houghton's "periodic message" as a heartbeat in view of Shen. For instance, Shen's *heartbeat message* was one well-known and widely implemented way to indicate the reachability of the device (in this case, to receive stored messages). Ex-1014, [0033]-[0034]. They were utilized so widely because they were known to be low-bandwidth (compared to transmitted messages themselves) and reliable mechanisms to indicate a device is connected and available for network transmissions (and in this case to trigger message delivery). *See, e.g.*, Ex-1043, [0062] (providing an example "64-byte link heartbeat (HB) message").

931. For these reasons, even if Houghton's periodic message cannot be considered a heartbeat message, in my opinion, replacing one with the other 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). In my opinion, a POSITA would have had a reasonable expectation of success in making this modification because Houghton and Shen already provide for similar signaling. *Id.*, [0034].

932. Upon making the combination, a POSITA would have understood that Houghton-Munson-Shen teaches a message-delivery trigger comprising *a heartbeat message* like Shen's, generated by the push client (*device link agent*).

> ***2.*** ***Claim 12:*** *The networked system of claim 1, the link interface to encrypt messages identified for delivery to each given one of the wireless end-user devices to create secure Internet data messages, the device link agent on each given device further configured to decrypt the received secure Internet data messages for that device prior to delivering those messages to a respective software process.*

933. Houghton-Munson-Shen renders obvious a link interface that encrypts messages identified for delivery to wireless devices to create a secure Internet data message, the device link agent on each wireless device configured to receive the message and decrypt it prior to delivering the messages to their respective software processes.

934. As described with reference to 1[b1] and 1[d4] (which I incorporate here), Houghton discloses delivering messages through secure push-server protocols (e.g., SSL and HTTPS) for delivery over a secure connection between the push server and the push client. Although Houghton discloses encrypting the connection between the push server and push client, it does not appear to disclose encrypting the messages themselves.

935. Shen, however, explains that the store and forward functionality like Houghton's (Ex-1007, 3, 7)—without encrypting the messages—exposes a "potential security problem" given that such systems are "not an end-to-end solution …." Ex-1014, [0004]. To overcome this "security problem," Shen's invention in one embodiment discloses that "an authentication and key management module 502 is added in the MMS gateway to distribute symmetric keys to users and a certificate management module 504 is added in the user device," as shown below (Ex-1014, [0054]):

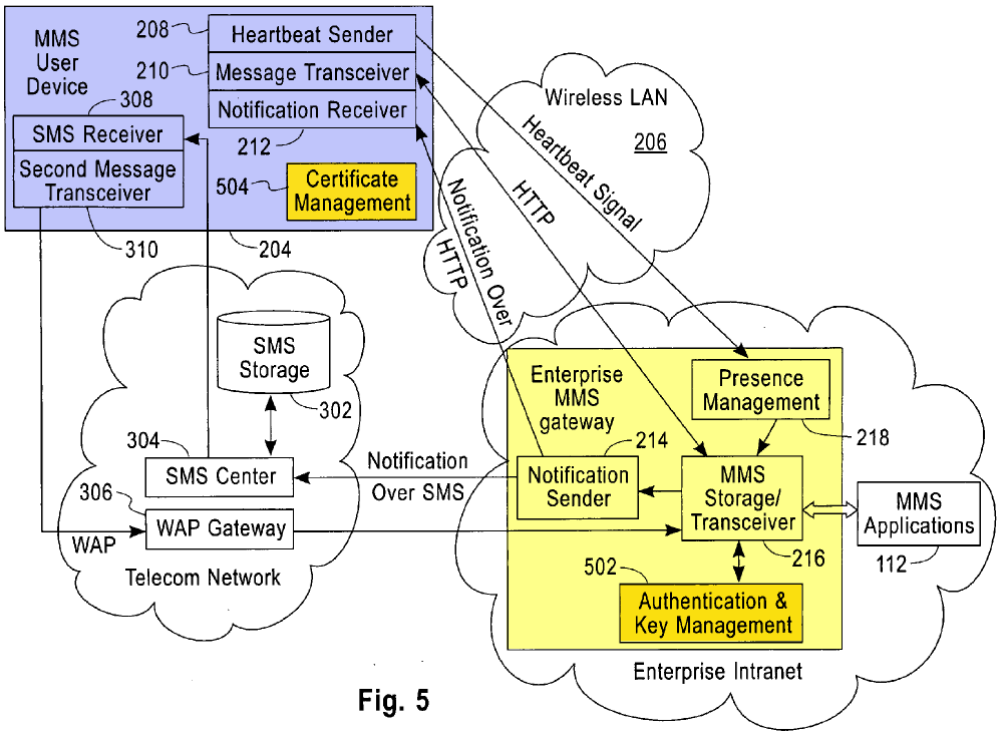


Fig. 5

Id., Fig. 5.

936. Using these modules, Shen discloses generation of a symmetric key that is then used by the MMS Relay/Server to encrypt data communications transmitted to the MMS User Device/Agent of user devices. *Id.*, [0055]-[0060]; *see also id.*, [0029]-[0034]. Shen discloses "encrypting MMS messages with the shared secret key." *Id.*, [0059]. Shen also discloses that messages can be "encode[d] and decode[d] to protect the privacy of these messages." *Id.*, [0054].

937. In my opinion, a POSITA would have been motivated to combine the teachings of Houghton-Munson and Shen to achieve a messaging system "having improved security," resulting from the "provi[sion] [of] an end-to-end security

solution for MMS applications," which is particularly beneficial in the context of "enterprise applications." Ex-1014, [0017], [0022] (describing additional advantages stemming from Shen's invention); Ex-1007, 22 (describing applicability of push messaging in the context of enterprise applications). Specifically, a POSITA would have been motivated to implement an "authentication and key management module 502" at the push server that generates and "distribute[s] symmetric keys to users" and using this symmetric key to encrypt data transmitted to "the user device," as contemplated by Shen. Ex-1014, [0054]-[0060].

938. Moreover, it was well known before the Critical Date for push servers to encrypt content of messages sent to push clients/client devices. Chou is one example of a push message system that uses a "WAP gateway" (e.g., a network message server) that encrypts content received from a "content server" for transmission to a "mobile client" that then "decodes the response." Ex-1009, 6:15-46. Rakic and Shen provide additional examples of a push server encrypting message traffic. Ex-1015, [0065]-[0068]; Ex-1014, [0054]-[0060]

939. Combining the above-described teaching of Shen with those of Houghton-Munson would have been obvious to a POSITA at least because such a combination would have merely involved combining prior art elements (a messaging network and message encryption) according to known methods (using encryption schemes and key generators) to yield predictable results (encrypted messages).

Similarly, the combination would have amounted to using a known technique (message encryption) to improve similar devices (messaging networks) in the same way (securing content).

940. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification because Shen describes applying encryption to a similar messaging environment as Houghton and communication between push servers and push clients, (Ex-1014, [0017]), such that a POSITA would have found straightforward to modify Houghton-Munson's push server to implement Shen's security modules.

941. Additionally, the resulting system's elements would each perform functions they were known to perform prior to the combination—i.e., Houghton-Munson's system would continue to facilitate communication of push messages between push servers and push clients, and Shen's teachings (as implemented in combination) would facilitate encryption by the push server for messages that are then sent to a push client.

942. In my opinion, upon making the combination, a POSITA would have understood that Houghton-Munson-Shen provides additional encryption protocols to encrypt messages identified for delivery supplied to the transport protocols for delivery on the secure link to encrypt messages, and thus create "secure internet data messages." Although I previously noted in 1[e3] (and which I incorporate here) that

use of secure protocols for the link forms "secure internet data messages," a POSITA would have appreciated that messages can be "secured" multiple ways, including more than one way simultaneously. In this implementation, for instance, a POSITA would have appreciated that not only is the message encrypted (rendering a message a "secure internet data message") but so is the entire link between, e.g., a push client and push server (also rendering the message a "secure internet data message"), meaning two layers of encryption would be applied in different parts of the stack of protocols.

943. And because it would have been obvious to provide encrypted messages, a POSITA would have further understood that each push client would be configured to decrypt the received messages prior to delivering those messages to their respective processes (e.g., either for display to a user, or to a registered application). If decryption were not provided at the receiver, any received (encrypted) messages would be unusable by the user/applications receiving them because encryption involves scrambling data into an unrecognizable form.

944. For the above reasons, Houghton-Munson-Shen renders obvious claim 12.

3. ***Claim 13:*** *The networked system of claim 12, wherein the encrypted messages are transported to the device link agent using one or more of encryption on a transport services stack, IP (Internet Protocol) layer encryption, and tunneling.*

945. Houghton-Munson-Shen renders obvious that the encrypted messages are transported to the device link agent on a wireless device using encryption on at least the transport services stack.

946. As I explained previously in 1[b1] (and incorporate here), Houghton discloses transmitting messages to push clients using "secure protocols" as part of, e.g., a transport stack. Moreover, as I explained in claim 12 (and incorporate here), the combination of Houghton, Munson, and Shen teaches encrypting transmitted messages to/from the push client.

947. For the above reasons, Houghton-Munson-Shen renders obvious claim 13.

4. ***Claim 16:*** *The networked system of claim 15, the device link agent on the given device further configured to buffer one or more of the upload messages for transmission to the network message server at a time selected by a heartbeat mechanism.*

948. As I described in claim 6 for both Ground 2A and 2D (which I incorporate here), Houghton discloses a process of testing a connection between a push client and push server using periodic messages, and Shen similarly teaches using a heartbeat mechanism for the client "to determine whether a wireless LAN is available." In other words, both Houghton and Shen teach a client device that tests a network connection to a server by sending a (heartbeat) message.

949. In my opinion, upon combining Houghton, Munson, and Shen, a POSITA would have been motivated to configure the push client to buffer messages

until the heartbeat mechanism determined it could reach the push server. It was well known that heartbeat messages typically required far less data than content messages—heartbeat messages could be on the order of bytes, *see, e.g.*, Ex-1043, [0062] (providing an example "64-byte link heartbeat (HB) message"), as opposed to kilobytes or even megabytes for content messages (to transmit text and multimedia messages such as images and video). A POSITA would have therefore been motivated to send a smaller amount of data to confirm the connection is still live *before* attempting transmission of the larger content messages. *See, e.g.*, Ex-1044, cl. 14 (describing "receiving data from the mobile device in the network, receipt of data from the mobile device occurring only after receipt of the connection confirm message in the mobile device.")

950. In my opinion, a POSITA would have had a reasonable expectation of success in making this modification because Houghton-Munson and Shen already provided all of the necessary software elements to carry out the modification—including heartbeat messages—and would simply require adjusting timers to trigger message delivery after the heartbeat confirmed the reachability of the push server.

951. For the above reasons, Houghton-Munson-Shen renders obvious claim 16.

L. Ground 2E: Claims 8 and 9 Are Rendered Obvious by Houghton, Munson, and Pazhyannur

*1. **Claims 8-9: [8]** The networked system of claim 1, further*

comprising a secure server to store a secure authorization list, the secure authorization list indicating the authorized software components and the authorized network elements that are allowed to communicate using the network server system; ***[9]*** *The networked system of claim 8, further comprising the respective device link agent on each wireless end-user device receiving access authorization information from the secure server, the access authorization indicating, respectively for each wireless end-user device, the software components authorized to receive messages via the device link agent on that device.*

952. Houghton-Munson-Pazhyannur renders obvious a secure server to store a secure authorization list to indicate authorized software elements (e.g., applications) and network elements (e.g., push clients or application servers) that are allowed to communicate (e.g., authorized) using the network server system (e.g., push server), where each device link agent (e.g., push client) receives access authorization information from the secure server indicating which software components (e.g., applications) are authorized to receive messages via the device link agent (e.g., push client) on that device.

953. As I explained previously in 1[b2] and 1[c2] (and incorporate here), a POSITA would have understood that applications in Houghton-Munson's system would register with the push client to transmit/receive messages. However, Houghton-Munson does not appear to disclose a (secure) server that maintains a list of registered network elements and applications.

954. Other prior art such as Pazhyannur discloses a network of user terminals and application servers. Ex-1037, Abstract. Pazhyannur recognizes that users may

register multiple devices associated with the user, and the system maintains a centralized user profile that lists all registered devices and applications to permit access to the applications on any of the user's registered devices. *Id.*, [0040].

955. In my opinion, a POSITA would have been motivated to implement a list of registered network elements (such as user terminals, application servers, etc.) and applications in, for instance, Houghton-Munson's push server to keep track ofall devices and applications associated with a user. *Id.*, [0040]. Moreover, it was well-known in the art that such lists provided the added benefit of controlling access to networks (for devices/applications on the list) to improve network security. *See, e.g.*, Ex-1008, [0044] ("the push authorization module 326 performs authorization of pushing a message to the client 310 based on a server white list which lists servers that are authorized to send a message to the client 310 through the gateway 320."), [0047].

956. In my opinion, a POSITA would have had a reasonable expectation of success in implementing a centralized list of registered network elements and applications because maintaining a centralized list of registered hardware and software simply requires compiling data the push clients and push servers would already have, such as the identifiers for registered applications or network-hardware connecting to the push server.

957. Upon making this modification, in my opinion, a POSITA would have appreciated that push clients and/or push servers would evaluate a list of registered network elements and applications on a server (like Houghton-Munson's push server) rather than a local list. That way, connections to the push server would be limited to already-approved hardware on the list, and applications would be manageable across different user devices.

958. For the above reasons, Houghton-Munson-Shen renders obvious claims 8 and 9.

M. Ground 2F: Claim 14 Is Rendered Obvious by Houghton, Munson, and Ellison

1. ***Claim 14:*** *The networked system of claim 1, wherein the device link agent on a given devices executes in a secure execution environment, and at least one of the software components on that device executes outside of the secure execution environment*

959. Houghton-Munson-Ellison renders obvious executing the device link agent (e.g., push client) in a secure execution environment, with applications that interact with the device link agent operated outside of the secure execution environment.

960. Houghton-Munson discloses both a push client (device link agent) and other applications executing on a mobile terminal. Ex-1007, 16, 21, Fig. 4. Houghton-Munson does not, however, appear to disclose implementing the push client in a secure environment.

961. In reference to Ground 1F (which I incorporate here), Ellison is just one example of prior art that teaches implementing software in a secure environment to protect it from potential attacks. Ex-1019, Abstract, 1:16-51. To inhibit attacks, Ellison discloses both an "isolated execution mode" (depicted in red below) where "access … is restricted" and a "normal execution mode" that "operates in a non-secure environment, or a normal environment" (depicted in green below) "without the security features provided by the isolated execution mode." Ex-1019, 4:65-5:1, 6:1-26, 8:25-32, Figs. 1A, 1C. Additionally, Ellison's teachings enable "normal" execution, for instance, for applications "***outside of the secure execution environment on that device***." *Id.*, 4:65-5:1, 6:1-26, Figs. 1A, 1C.

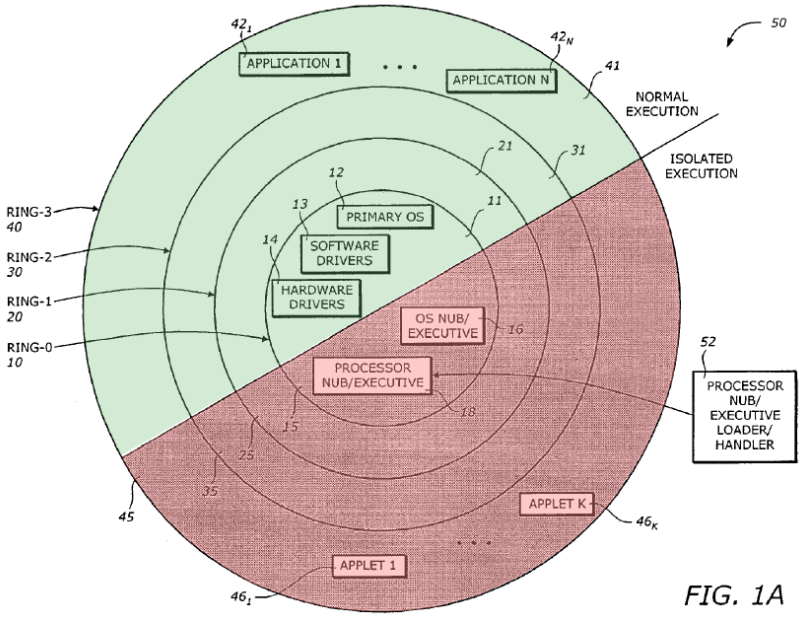


FIG. 1A

Id., Fig. 1A.

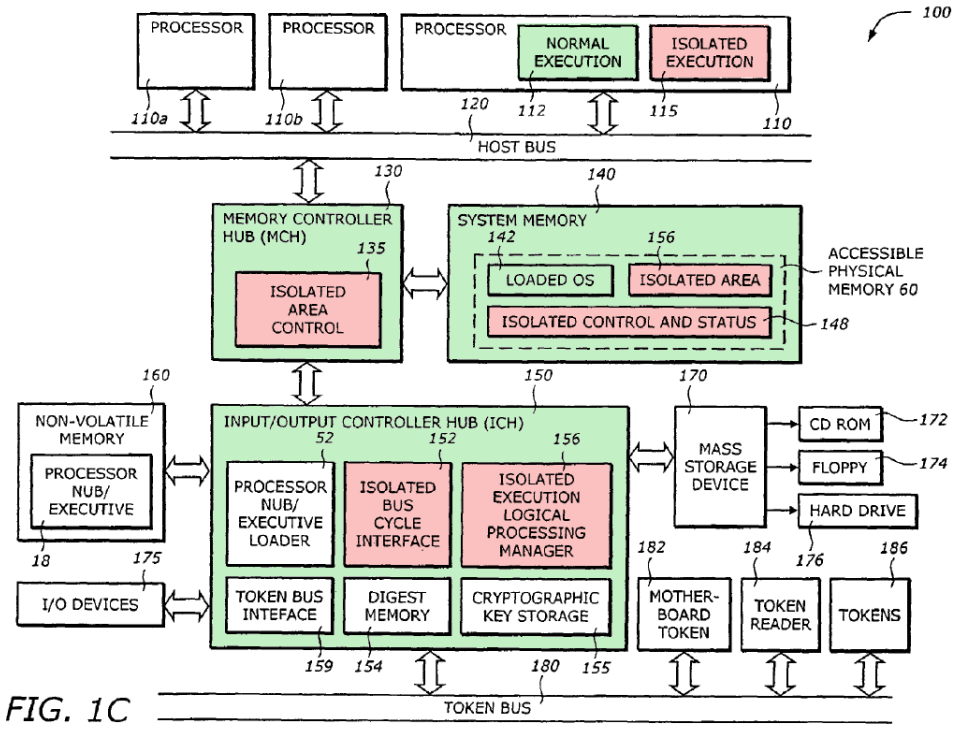


FIG. 1C

Id., Fig. 1C.

962. Ellison further discloses an "isolated area 70" which is "[a] memory area that is defined by the processor 110 when operating in [an] isolated execution mode." Ex-1019, 6:1-26, 8:25-32, Figs. 1A-1C, 2. Ellison refers to this technique, illustrated below in Figure 2, as a "secure platform." Ex-1019, 6:1-26, 8:25-32, 8:66-9:62, Figs. 1A-1C, 2.

963. Ellison further teaches that the secure platform includes a "key generator 240" that "generates a key operating system nub key (OSNK) 203" which is then supplied to "trusted agents." *Id.*, 8:66-9:6, Fig. 2. One example of a trusted agent is a "usage protector 250" that "uses the OSNK 203 to protect the usage of [a] subset 230." *Id.*, 9:28-40, Fig. 2. Ellison's usage protector 250 uses a hashing

function with the subset 203's ONSK 203 to determine if the subset 203 has been altered following changes (e.g., reads or writes to the subset 203), thereby providing "protection against unauthorized reads, and detection of intrusion, tampering or unauthorized modification." *Id.*, 9:47-62.

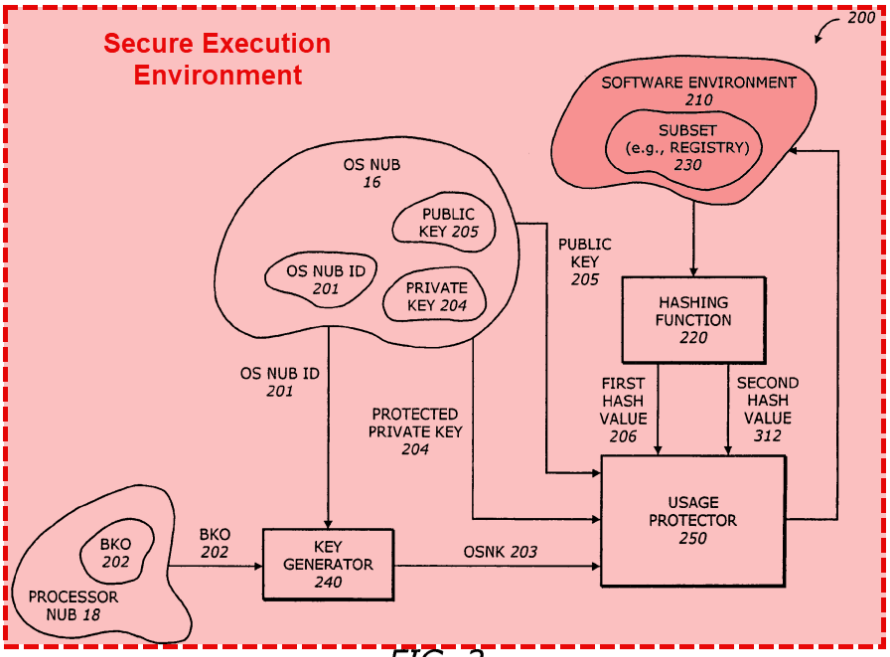


FIG. 2

Id., Fig. 2.

964. In my opinion, a POSITA would have been motivated to combine the teachings of Houghton-Munson and Ellison (and specifically to incorporate Ellison's hierarchical, nub-based normal and isolated execution environments into Houghton-Munson) to improve the security of Houghton's mobile terminal. *See, e.g.*, Ex-1007, Figs. 3-4, 8; Ex-1019, 4:63-5:10, 8:25-32, 8:66-9:6, Figs. 1A, 2. Such a combination would ensure that malicious applications could not easily gain access

to stored message content after the encryption used during transmission had been removed by the intended user device.

965. A POSITA would have recognized that Ellison's secure platform would have improved the security of Houghton's push message system by improving device security (i.e., entry points to the network). Ex-1019, 8:25-32, 8:66-9:6, 9:28-62, Fig. 2.

966. The increased confidence in device and/or network security provided by Ellison would have enabled a POSITA to Houghton's push service to additional application servers and clients. *Id.*, 8:25-32, 8:66-9:6, 9:28-62, Fig. 2.

967. Moreover, device level security improvements are an efficient, beneficial option for improving network security, which would have been a natural solution for Houghton's system, which is concerned with secure messaging in the push messaging context. *Id.*, 8:25-32, 8:66-9:6, 9:28-62, Fig. 2.

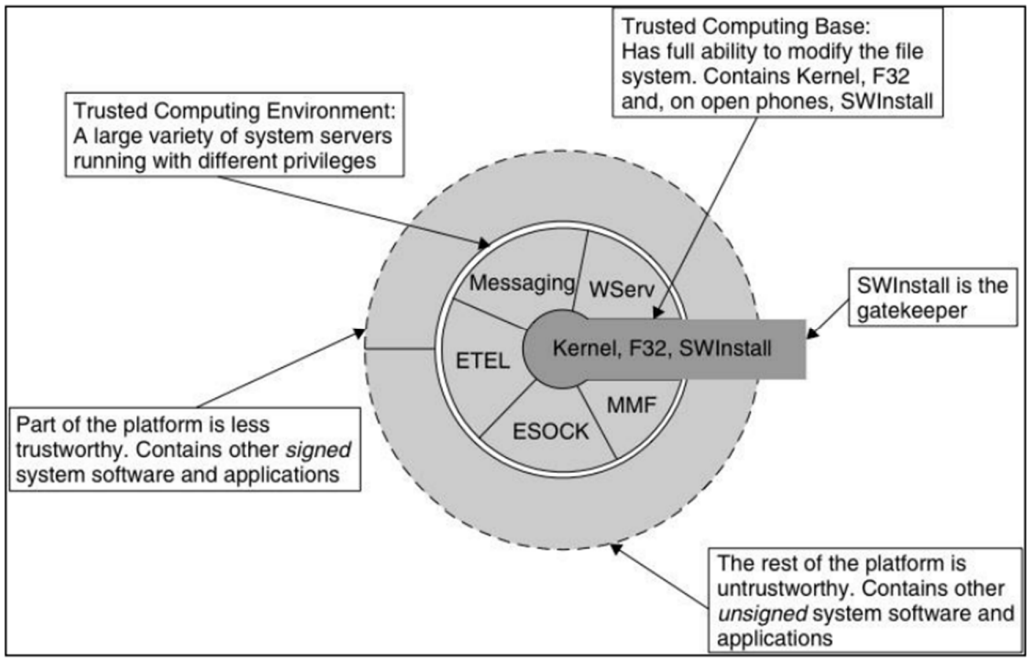
968. A POSITA would have had a reasonable expectation of success in combining Houghton-Munson and Ellison because (1) Houghton contemplates security mechanisms to secure its messaging systems, including messaging around application data; (2) Ellison's techniques can be implemented in "computer system[s]" similar to the network described in the Houghton-Munson; and (3) it was well-known to use protection rings and tiers in computing environments (per Ellison). Ex-1019, 2:41-3:31, Fig. 1A.

969. I understand that combining the teachings of Houghton, Munson, and Ellison would have been obvious at least because such a combination would have merely involved combining prior art elements (computer messaging networks and secure computer executing environments) according to known methods (software/hardware implementations) to yield predictable results (select computer programs executing in secured environments).

970. In my opinion, a POSITA would have also had a reasonable expectation of success in incorporating Ellison's security techniques into Houghton's mobile terminals because Ellison describes that its techniques can be implemented in "computer system[s]" that contain a "processor" similar to the network described in Houghton. *Id.*, 5:11-16. A POSITA would have recognized or found it obvious that the examples of "mobile terminals" provided in Houghton include a "processor" capable of implementing Ellison's techniques as it was known in the art that mobile terminals contained processors. Ex-1007, 16; *see also* Ex-1019, 8:25-32, 8:66-9:6, 9:28-62, Fig. 2.

971. Moreover, in the context of mobile devices and their associated operating systems, it was well-known before the Critical Date to implement more secure systems, with rings or tiers of protection around different device components. *Id.*, 2:41-3:31, Fig. 1A. For example, in the early 2000s, the Symbian mobile operating system (B. Li, et al., *Symbian OS platform security model*, LOGIN:

Magazine (Aug. 2010)—Ex-1033; Simon Higginson, Platform Security Concepts, in SYMBIAN OS PLATFORM SECURITY: SOFTWARE DEVELOPMENT USING THE SYMBIAN OS SECURITY ARCHITECTURE, 17-41 (Craig Heath ed., 2006)—Ex-1045) separated processes into three "Trust Tiers": The Trusted Computing Base (TCB), The Trusted Computing Environment (TCE), and Applications, as illustrated below:



Ex-1045, 24 (Fig. 1)

972. As shown above, the first two layers/tiers shown were primarily dedicated to components of the Operating System with relatively higher operating privileged, followed by the Applications tier with applications having relatively lower privileges. Ex-1033, 3-5, Fig. 1. The first two tiers included the Trusted Computing Base and the Trusted Computing Environment. *Id.*, Fig. 1. As shown,

the Trusted Computing Environment includes, among other services, a Messaging service. *Id.* The Application Tier could be further subdivided into Signed and Unsigned applications. *Id.* Signed applications could interact with the TCE and have this layer execute higher privileged requests, whereas Unsigned applications could not access that same functionality. *Id.*, 4, Fig. 4.

973. Thus, it was well-known for operating systems to include the messaging services and clients/agents (e.g., push clients) responsible for the provision of those services, within the trusted computing environment, and for applications (including those that would be intended recipients of those messages) to be tiered/organized outside this environment.

974. Given that Houghton contemplates mobile terminals with a push client that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging) from other applications/components of the environment, and Ellison's own disclosures confirming the same, a POSITA would have been motivated to similarly enable such improved security within the push messaging environment in Houghton by implementing Ellison's teachings of secure environments, such that the push client is part of the device's secure execution environment and enables communications with applications that reside outside this trusted/secure execution environment. Such a combination would have been straightforward to implement

with a reasonable expectation of success given the well-known use of such techniques in the art and the recognition and use of security/encryption already contemplated in Houghton.

975. Specifically, a POSITA would have understood that Houghton's push client would operate in an isolated, secure environment that "is protected by both the processor and chipset" (Ex-1019, 3:32-48), to reduce the push client/user's messages exposure to attack. Applications that interact with the push client would operate in normal or separate isolated environments to restrict the application(s) access to the push client. *Id.*, 2:55-61 (noting applications operate in "Ring-3"—"the lowest level of the hierarchy" with the "least privilege").

976. In this manner, Houghton-Munson-Ellison renders obvious that the push client executes in a secure execution environment and one or more of the applications to which push messages are routed executes outside of the secure execution environment.

N. Ground 2G: Claim 17 Is Rendered Obvious by Houghton, Munson, and Fok

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

977. Houghton-Munson-Fok renders obvious a secure interprocess communication service that is separately secured from the secure Internet data message link, with a device link agent (e.g., push client) on a given device causing messages to be securely delivered to a software process (e.g., application process) by initiating delivery on the secure interprocess communication device.

978. As discussed above in 1[b2] (and incorporated by reference here), Houghton discloses transmitting application-specific messages across a network. However, because Houghton-Munson does not expressly describe inter-application transmissions on a given device.

979. Fok describes a mobile environment that includes a secure inter-application communication platform. Ex-1038, [0036]-[0038], [0081], [0106]-[0108], Figs. 4, 8-9. Fok uses a "handshake" between a "primary application" and one or more "recipient" applications to confirm that they are trusted application using "a list of unique identifiers" before transmitting information between the two (causing messages to be securely delivered to a software process). *Id.*, [0047]-[0053], [0096]-[0102], [0106]-[0108], Figs. 7-10.

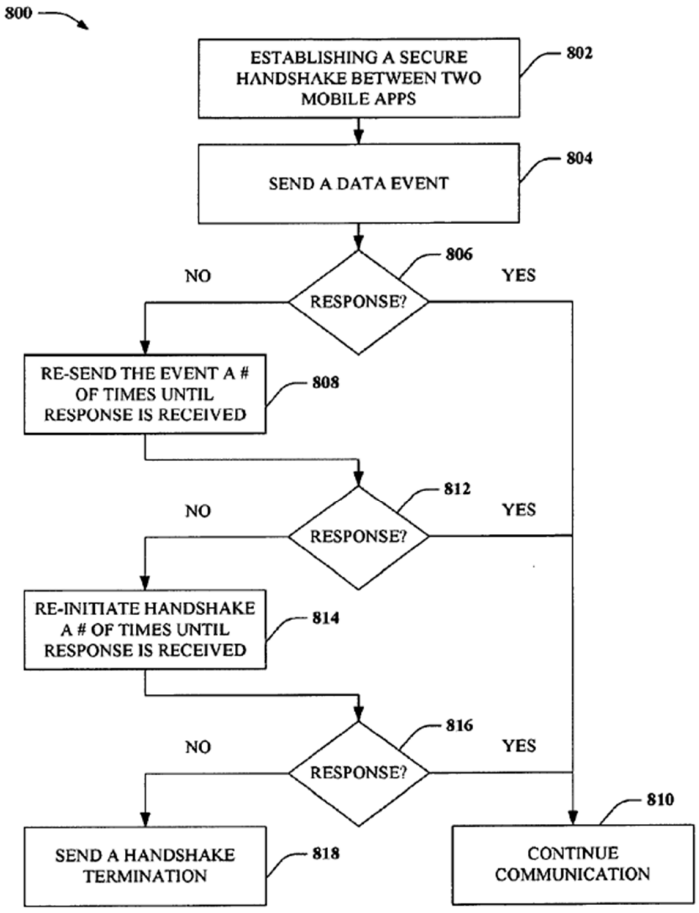


FIG. 8

Id., Fig. 8

980. In my opinion, a POSITA would have been motivated to combine Houghton-Munson and Fok to provide a "secure inter-application communication" like Fok's so that two or more applications residing on a single UE/wireless device may transmit application messages to one another through the push client on the terminal. That way, applications on a single user device would be able to interact with one another without needing to develop, for instance, custom APIs ("Application Programming Interfaces") for alternative forms of direct

communication, and ensure the applications were trusted before sharing information; the push client would provide the interface and relevant authorizations. As one example, a messaging application could be configured to forward user-specified messages with images to a photo-storage or editing application on the same device. Or a messaging application could direct a video-playback application to download a specific codec to play back a video message in the video-playback application.

981. A POSITA would have understood that this inter-application communication would be secured separately from the link established between the push client and push server because inter-application messaging does not need to utilize an internet network or reach the push server when transmitting messages between different applications on a single user terminal (since both applications exist on a single device). A POSITA would have appreciated that this separate inter-application channel would also be "secured" to prevent unauthorized access to the message transmissions between the two applications. *See, e.g.*, *id.*, [0036]-[0038]. Moreover, although Fok discloses a direct handshake between the two applications, a POSITA would have appreciated that the combination of Houghton-Munson and Fok would instead pass through the push client (so one application sends the push client a message, which the push client then forwards to the receiving application). That way, each application would not need to be configured with a brand-new

"handshake" mechanism, but would instead utilize its existing protocols for Houghton-Munson's transmissions.

982. Moreover, combining Houghton-Munson and Fok's teachings would have been obvious because such a combination would have merely involved combining prior art elements according to known methods to yield predictable results. Incorporating Fok's inter-application communication techniques into Houghton-Munson's mobile terminals would also have been predictable and foreseeable with a reasonable expectation of success because Fok describes that its techniques are directed towards mobile operating environments, similar to Houghton-Munson's terminals and push client., *Id.*, [0002].

983. In my opinion, in reference to Ground 1G (which I incorporate here), a POSITA would have had a reasonable expectation of success in making this combination because Fok already describes how to establish an inter-application connection; the only difference is the connection would be established through Houghton-Munson's push client. Moreover, this internal-facing, device-specific connection would only supplement, and not replace, Houghton-Munson's external-facing, networked connection.

XI. ADDITIONAL REMARKS

984. I currently hold the opinions expressed in this declaration. But my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.

985. I hereby declare that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I further declare that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of the Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Dated: 10 November 2025 By: 

Patrick Gerard Traynor

Professor
Associate Chair for Research in CISE
John and Mary Lou Dasburgh Preeminent Chair in Engineering

Department of Computer & Information Science & Engineering (CISE)
University of Florida
1889 Museum Rd,
Gainesville, FL 32611 USA
traynor@cise.ufl.edu
http://www.cise.ufl.edu/~traynor

Table of Contents

EDUCATIONAL BACKGROUND 4

EMPLOYMENT HISTORY 4

CURRENT FIELDS OF INTEREST 4

I. TEACHING 6
- A. Courses Taught . . . 6
- B. Continuing Education . . . 6
- C. Curriculum Development . . . 6
- D. Individual Student Guidance . . . 7
- E. Teaching Honors and Awards . . . 12

II. RESEARCH AND CREATIVE SCHOLARSHIP 13
- A. Thesis . . . 13
- B. Published Journal Papers (Refereed) . . . 13
- C. Published Books and Parts of Books . . . 15
- D. Edited Proceedings . . . 15
- E. Conference Presentations . . . 15
  - E.1. Conference Papers with Proceedings (Refereed) . . . 15
  - E.2. Conference Presentations with Proceedings (Non-Refereed) . . . 23
  - E.3. Conference Presentations without Proceedings . . . 23
- F. Other . . . 23
  - F.1. Submitted Journal Papers . . . 23
  - F.2. Refereed Research Reports . . . 23
  - F.3. Software . . . 23
  - F.4. Published Papers (Non-Refereed) . . . 23
  - F.5. Books in Preparation . . . 23
  - F.6. Workshops and External Courses . . . 24
- G. Research Proposals and Grants (Principal Investigator) . . . 24
- H. Research Proposals and Grants (Contributor) . . . 27
- I. Research Honors and Awards . . . 28

III. SERVICE 29
- A. Professional Activities . . . 29
  - A.1. Memberships and Activities in Professional Societies . . . 29
  - A.2. Conference Committee Activities . . . 29
- B. On-Campus Committees . . . 30
  - B.1. University of Florida . . . 30
  - B.2. Georgia Tech . . . 31
- C. Special Assignments . . . 31
- D. Ph.D. Examining Committees . . . 31
- E. External Member of M.S. Examining Committee . . . 35
- F. Consulting and Advisory Appointments . . . 35
- G. Civic Activities . . . 35

IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION 36
- A. Honors and Awards . . . 36
- B. Invited Conference Session Chairmanships . . . 36
- C. Professional Registration . . . 36
- D. Patents . . . 36
- E. Editorial and Reviewer Work for Technical Journals and Publishers . . . 37
- F. Expert Witness Services . . . 39

V. OTHER CONTRIBUTIONS 41
- A. Seminar Presentations (Invited Papers and Talks at Meetings and Symposia) . . . 41

B. Special Activities 45

EDUCATIONAL BACKGROUND

Degree	Year	University	Field
Ph.D.	2008	Pennsylvania State University State College, PA *Dissertation:* Characterizing the Impact of Ridigity on the Security of Cellular Telecommunications Networks *Advisors:* Thomas F. La Porta and Patrick D. McDaniel	Computer Science & Engineering
M.S.	2004	Pennsylvania State University State College, PA	Computer Science & Engineering
B.S.	2002	University of Richmond Richmond, VA *Minors:* Biology, Business Admin	Computer Science

EMPLOYMENT HISTORY

Title	Organization	Years
Interim Department Chair	University of Florida	*July 2025–Present*
Associate Chair for Research	University of Florida	*August 2018–Present*
Professor	University of Florida	*August 2018–Present*
Associate Professor	University of Florida	*August 2014–July 2018*
Associate Professor	Georgia Institute of Technology	*March 2014–August 2014*
Assistant Professor	Georgia Institute of Technology	*2008–March 2014*
Research Assistant	Pennsylvania State University	*2004–2008*
Teaching Assistant	Pennsylvania State University	*2004*

CURRENT FIELDS OF INTEREST

My research focuses on the security of cellular/telephony networks and mobile systems. The security of these systems generally relies on their closed nature and trust in the honest behavior of users. However, with the recent disintegration of these assumptions and with over than six billion subscribers around the world, cellular and mobile systems represent the next great expansion in global critical infrastructure and, because of their unique characteristics, require new and different approaches to security.

Recognizing this, my research focuses on three specific themes: (1) developing efficient techniques to allow telephony providers and customers to authenticate the origin of incoming calls; (2) measuring and

improving the security of emerging mobile financial systems and (3) efficient and strong privacy-preserving techniques for mobile devices. Additionally, I have significant expertise in fraud detection, particularly for payment systems.

I have a strong interest in solutions that can be deployed in both the short and long terms, and am actively engaging both industry and government in this capacity. My research, if successful, will help to not only improve the general security of networked devices, but also to maintain the historical reliability of telephony networks as they become the dominant digital access technology.

I. TEACHING

A. Courses Taught

Semester/Year	Course Number & Title	Number of Students	Comments
Fall 2024	CNT 4007 Computer Networks 1	310	Revamped Course
Fall 2023	CIS 6930 Cellular and Mobile Network Security	19	New Topics
Fall 2022	CNT 5410 Computer and Network Security	75	New Topics
Fall 2021	CNT 5410 Computer and Network Security	45	New Topics
Fall 2019	CNT 5410 Computer and Network Security	28	New Topics
Fall 2018	CIS 6930 Cellular and Mobile Network Security	16	New Course
Fall 2017	CNT 5410 Computer and Network Security	27	New Topics
Fall 2016	CNT 5410 Computer and Network Security	60	New Topics
Spring 2016	CNT 5410 Computer and Network Security	13	New Topics
Spring 2015	CNT 5410 Computer and Network Security	12	New Topics
Fall 2014	CNT 5410 Computer and Network Security	30	New Course
Spring 2014	CS 6262 Network Security	55	New Projects
Fall 2013	CS 3251 Computer Networks I	73	Expanded Syllabus
Spring 2013	CS 6262 Network Security	65	All New Projects
	CS 8001 Information Security Seminar	20	New Speakers
Fall 2012	CS 8803 Cellular & Mobile Network Security	17	New Topics
	CS 8001 Information Security Seminar	20	New Speakers
Spring 2011	CS 8001 Information Security Seminar	20	New Speakers
Fall 2011	CS 6262 Network Security	27	Expanded Syllabus
	CS 8001 Information Security Seminar	35	New Speakers
Spring 2011	CS 3251 Computer Networks I	61	Expanded Syllabus
	CS 8001 Information Security Seminar	20	New Speakers
Fall 2010	CS 8803/4803 Cellular & Mobile Network Security	16	New Course
	CS 8001 Information Security Seminar	31	New Speakers
Fall 2009	CS 6262 Network Security	55	Expanded Syllabus
Spring 2009	CS 3251 Computer Networks I	45	Expanded Syllabus
Fall 2008	CS 8003 Destructive Research	10	New Course

Guest lecturer for CS 4235 (Introduction to Information Security) and CS 8803 (e-Democracy) in Fall 2008.

Advised ECE 4811/CS 4802 (Vertically Integrated Project) with Ed Coyle

B. Continuing Education

None.

C. Curriculum Development

University of Florida

CNT 4007 Computer Networks 1: *Fall 2024.* Provided the first major overhaul to this course in a number of years. While I have relied on the same book used by other faculty, I have created new homeworks, projects, and slides to better represent the current state of computer networks. I have also significantly expanded the discussion of security in this course.

CIS 6930 Cellular and Mobile Network Security: *Fall 2018, 2023.* Developed an entirely new course around security issues facing cellular and mobile networks. Students learned about wireless basics, spectrum issues, core network architectures (GSM, ISDN, IMS, SIP), air interfaces (2G-5G), mobility management, authentication, mobile phone operating systems (Android, iPhone), Android security, congestion and denial of service, privacy and eavesdropping. Students also complete a research project and aim towards publishing this work at a major venue. My aim is for this class to become part of the regular offering of security courses. Semester projects were also judged and encouraged using a "venture capital" model, in which students had to pretend as if they were pitching their ideas for a start-up company to potential investors.

CNT 5410 Computer and Network Security: *Fall 2014-2022.* Totally rewrote the syllabus and slide material, giving the class its first major overhaul in a number of years. While many old themes remain, new lecture blocks including Web Security, Cellular Security and Social Engineering were developed from scratch. This new course material was made available to all other faculty members teaching this class, who have since used my slides and syllabus.

Georgia Tech In addition to the above courses, I also developed the following course while serving as a faculty member at Georgia Tech.

CS 3251 Computer Networks I: *Spring 2009.* Modified undergraduate networking course to include a persistent focus on security at all layers of the protocol stack. I have also created new lectures focusing on the physical layer and cellular networks and new exams to include all of the abovementioned changes.

CS 8803 Destructive Research: *Fall 2008.* Developed course based around understanding how so-called secure systems have been defeated by attackers. With such knowledge, students would have the context to develop the next generation of more secure systems. I delivered more than 1/3 of the lectures in this seminar course and paid special focus on vulnerabilities in cellular networks, analog telecommunications and electronic voting. Students were also instructed on techniques for performing research, writing technical papers and making conference and lecture-style presentations. I have offered these slides to future 7001 classes to help impact a wider audience.

D. Individual Student Guidance

1. Research Scientists Supervised

None.

2. Ph.D. Students Graduated

Hadi Abdullah University of Florida
Fall 2016–Summer 2022
Evaluating the security of ML-driven voice interfaces. Now: Research Scientist at Visa Research

Chaitrali Amrutkar Georgia Institute of Technology
Fall 2009–Fall 2013
Her research discovered vulnerabilities in mobile web browsers and developed techniques to detect malicious mobile web pages. Joined Oracle in Spring 2014.

Logan Blue University of Florida
Fall 2016–Summer 2022
Investigated biological feature reconstruction from voice recordings. Now: Research Scientist at Harbor Labs

Jasmine Bowers University of Florida
Fall 2015–Summer 2020
Her research focuses on mobile applications, and the development of tools for building secure systems. Now: Research Scientist, MITRE

Henry "Hank" Carter Georgia Institute of Technology
Fall 2010–Spring 2016
Developing techniques for secure function evaluation for privacy-preserving applications on constrained mobile devices. Now: Assistant Professor, Villanova University

Italo Dacosta Georgia Institute of Technology
Fall 2008–Summer 2012
Co-advised with Mustaque Ahamad. Research on scaling performance of SIP network components. Graduated Summer 2012, currently research scientist at EPFL.

David Dewey Georgia Institute of Technology
Fall 2011–Summer 2015
Investigated compiler techniques to remove software vulnerabilities from code. Now CTO of MailChimp.

Cassidy Gibson University of Florida
Fall 2019–Spring 2025
Investigated the use of AI in generating non-consensual imagery. Now Research Scientist at ActiveFence.

Seth Layton University of Florida
Fall 2020–Summer 2025
Detecting deepfakes in audio samples. Now Research Scientist at PinDrop.

Christian Peeters University of Florida
Fall 2016–Summer 2022
Develop techniques to detect and defend against call and message interception attacks in cellular networks. Now: Research Scientist at Harbor Labs

Brad Reaves University of Florida
Fall 2014–Spring 2017
Develop strong authentication techniques for cellular networks. Now: Assistant Professor at North Carolina State University.

Nolen Scaife University of Florida
Fall 2014–Spring 2019
Developed techniques to detect credit card skimming. First: Assistant Professor at the University of Colorado Boulder. Now: Director, Global Cyber Intelligence at Walmart

Imani Sherman University of Florida
Fall 2018–Summer 2021
Developing usable interfaces against robocalls. Co-advised with Juan Gilbert. Now: Assistant Professor at the University of California, San Diego

Tyler Tucker University of Florida
Fall 2021–Summer 2025
Evaluating the security of Bluetooth/cellular radios. Now: Associate at the Analysis Group.

Luis Vargas University of Florida
Fall 2016–Summer 2021
Developing techniques for network-based detection and mitigation of malware in a healthcare environment. Now: Data Scientist at the Alethia Group

Kevin Warren University of Florida
Fall 2019–Summer 2025
Detecting deepfake audio through linguistic information. Now: Associate at the Analysis Group.

2. Ph.D. Students Supervised

Nathaniel Bennett University of Florida
Fall 2022–Present
Finding vulnerabilities in cellular core networks via fuzzing.

Jordan Greene University of Florida
Fall 2025–Present
Cellular network security.

Allison Lu University of Florida
Fall 2022–Present
Measuring repeatability in computer security.

Daniel Olszewski University of Florida
Fall 2019–Present
Removing unwanted/insecure features from software.

Aviva Smith University of Florida
Fall 2025–Present
Detecting Deepfakes.

3. Ph.D. Students - Other

Saurabh Chakradeo Georgia Institute of Technology
Fall 2010–Spring 2013
Research exploring malicious mobile applications. Left to join Facebook.

Brendan Dolan-Gavitt Georgia Institute of Technology
Spring 2009
Research project on using kernel type graphs to detect dummy structures.

Ryon Kennedy University of Florida
Fall 2020–Spring 2023
Finding vulnerabilities in cellular core networks via fuzzing. Left to join UFIT.

Eric (Yu) Liu Georgia Institute of Technology
Fall 2008
Research on the spread of malcode through cellular infrastructure.

Chaz Lever Georgia Institute of Technology
Fall 2011–Spring 2014
Developing techniques to measure the spread of malware in cellular networks. Left Georgia Tech to create a startup.

Frank Park Georgia Institute of Technology
Fall 2008–Spring 2010
Research on multi-factor authentication using cellular phones. Left program after failing comprehensive exam to join startup.

Ferdinand Schober Georgia Institute of Technology
Fall 2009–Summer 2010
Developed mechanisms for smart networks and smart mobile devices to fight infection and provide remote remediation. Returned to Microsoft.

4. M.S. Students Supervised

Chaitrali Amrutkar Georgia Institute of Technology
Fall 2008–Spring 2009
Research on improving performance of security critical functions in IMS cellular core. Completed her Ph.D with me at GT.

Logan Blue University of Florida
Fall 2015–Spring 2016
Investigated problems of cellular and network security.

David Dewey Georgia Institute of Technology
Fall 2009–Spring 2010
Research on security issues caused by transitive trust assumptions in the Windows COM infrastructure. Completed his Ph.D. with me at GT.

Christopher Grayson Georgia Institute of Technology
Fall 2012–Fall 2013
Developed continuous authentication mechanisms using the multitude of sensors available on a mobile phone. Now at Bishop Fox Consulting (industry).

Young Seuk Kim Georgia Institute of Technology
Fall 2012–Fall 2013
Performed research that compared the security vulnerabilities found in the traditional and mobile web.

Daniel Komaromy Georgia Institute of Technology
Fall 2008–Summer 2009
Research on building a real-time streaming audio system using attribute-based crypto for broadcast encryption.

Nigel Lawrence Georgia Institute of Technology
Fall 2011–Spring 2012
Discovered hijacking attacks in SNMPv3, a widely used and thought to be secure network management protocol. Now at Solute (industry).

Philip Marquardt Georgia Institute of Technology
Fall 2009–Present
Research on developing an iPhone application to prevent individuals from being profiled by Shopper Loyalty Programs. First with MIT Lincoln Labs, now Raytheon

Rishikesh Naik Georgia Institute of Technology
Fall 2008–Spring 2010
Research on converting expensive cryptographic primitives (e.g., Secure Function Evaluation) into efficient applications for mobile phones. Now with Cisco Systems.

Ashish Nautiyal University of Florida
Fall 2015–Spring 2016
Research on connecting telephone calls to the larger authentication infrastructure.

Nilesh Nipane Georgia Institute of Technology
Fall 2008–Spring 2010
Research on creating provably anonymous networks on a base of secure function evaluation. Now with VMWare.

Walter "Nolen" Sciafe Georgia Institute of Technology
Spring 2012–Spring 2014
Developed the OnionDNS architecture, which prevents domain delisting attacks by leveraging a Tor hidden service. Joined Ph.D. program at UF.

Tyler Tucker University of Florida
Fall 2018–Spring 2021
Evaluating the security of Bluetooth radios.

5. M.S. Special Problems Students

Siddhant Deshmukh University of Florida
Fall 2016–Present
Developed tools for analysis of mobile digital financial services.

Chinmay Gangakhedkar Georgia Institute of Technology
Spring 2009
Research on multi-factor authentication using mobile phones.

Christopher Grayson Georgia Institute of Technology
Spring 2013
Research on continuous authentication using mobile phones.

Aarushi Karnany University of Florida
Fall 2016–Present
Developed tools for analysis of mobile digital financial services.

Rohit Matthews Georgia Institute of Technology
Spring 2011
Developed mobile phone-based tools for measuring performance and reachability throughout the Internet.

Ashwin Narasimhan Georgia Institute of Technology
Spring 2009
Research on developing efficient security mechanisms for the IMS cellular core.

Aamir Poonawalla Georgia Institute of Technology
Spring 2010
Helped develop a call provenance infrastructure, which included both networking and machine learning components.

Erin Reddick Georgia Institute of Technology
Fall 2008–Fall 2009
Research on IPTV security with GTRI.

Lalanthika Vasudevan Georgia Institute of Technology
Spring 2009
Research on developing efficient security mechanisms for the IMS cellular core.

6. Undergraduate Special Problems Students

Ethan Shernan Georgia Institute of Technology
Spring 2014
Developed an infrastructure for detecting billing bypass fraud attacks.

Young Seuk Kim Georgia Institute of Technology
Fall 2011–Spring 2012
Developed a mobile phone application for taking measurements of cellular networks.

Dane Van Dyck Georgia Institute of Technology
Summer 2009
Research on virtualization support for mobile phones.

E. Teaching Honors and Awards

1. Georgia Institute of Technology, CETL "Thanks For Being A Great Teacher" Award, Fall 2013.
2. Georgia Institute of Technology, CETL "Thanks For Being A Great Teacher" Award, Fall 2012.
3. United State Army Signal Corps, "Helmet" Award, 2010.
4. Georgia Institute of Technology, CETL "Thanks For Being A Great Teacher" Award, Spring 2009.
5. Pennsylvania State University CSE Graduate Student Teaching Award, 2005

II. RESEARCH AND CREATIVE SCHOLARSHIP

A. Thesis

1. Patrick Gerard Traynor. *Characterizing the Impact of Rigidity on the Security of Cellular Telecommunications Networks*. PhD thesis, The Pennsylvania State University, May 2008.

B. Published Journal Papers (Refereed)

1. Cassidy Gibson, Vanessa Frost, Katie Platt, Washington Garcia, Luis Vargas, Sara Rampazzi, Vincent Bindschaedler, Patrick Traynor, and Kevin Butler. Analyzing the Monetization Ecosystem of Stalkerware. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2022. (Acceptance rate: 24%).

2. Bradley Reaves, Luis Vargas, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Characterizing the Security of the SMS Ecosystem with Public Gateways. *ACM Transactions on Privacy and Security (TOPS)*, 22(1), 2018.

3. Patrick Traynor, Kevin Butler, Jasmine Bowers, and Bradley Reaves. FinTechSec: Addressing the Scurity Challenges of Digital Financial Services. *IEEE S&P Magazine*, 15(5):85–89, 2017.

4. Nolen Sciafe, Henry Carter, Rachel Jones, Lyrissa Lidsky, and Patrick Traynor. OnionDNS: A Seizure-Resistant Top-level Domain. *International Journal of Information Security (IJIS)*, 2017.

5. Bradley Reaves, Jasmine Bowers, Nolen Scaife, Adam Bates, Arnav Bharatiya, Patrick Traynor, and Kevin Butler. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. *ACM Transactions on Privacy and Seurity (TOPS)*, 2017.

6. Henry Carter and Patrick Traynor. OPFE: Outsourcing Computation for Private Function Evaluation. *International Journal of Information and Computer Security (IJICS)*, 2017.

7. Stephan Heuser, Bradley Reaves, Praveen Kumar Pendyala, Henry Carter, Alexandra Dmitrienko, William Enck, Negar Kiyavash, Ahmad-Reza Sadeghi, and Patrick Traynor. Phonion: Practical Protection of Metadata in Telephony Networks. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2017.

8. Bradley Reaves, Jasmine Bowers, Sigmond A. Gorski III, Olabode Anise, Rahul Bobhate, Raymond Cho, Hiranava Das, Sharique Hussain, Hamza Karachiwala, Nolen Scaife, Byron Wright, Kevin Butler, William Enck, and Patrick Traynor. *droid: Assessment and Evaluation of Android Application Analysis Tools. *ACM Computing Surveys (CSUR)*, 2016.

9. Chaitrali Amrutkar, Young Seuk Kim, and Patrick Traynor. Detecting Mobile Malicious Webpages in Real Time. *IEEE Transactions on Mobile Computing (TMC)*, To Appear 2016.

10. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Outsourcing Secure Two-Party Computation as a Black Box. *Journal of Security and Communication Networks (SCN)*, To Appear 2016.

11. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Secure Outsourced Garbled Circuit Evaluation for Mobile Devices. *Journal of Computer Security (JCS)*, 24(2):137–180, 2016.

12. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. Accountable Wiretapping -or- I Know They Can Hear You Now. *Journal of Computer Security (JCS)*, 23(2):167–195, 2015.

13. Henry Carter, Chaitrali Amrutkar, Italo Dacosta, and Patrick Traynor. For Your Phone Only: Custom Protocols for Efficient Secure Function Evaluation on Mobile Devices. *Journal of Security and Communication Networks (SCN)*, 7(7):1165–1176, 2014.

14. Chaitrali Amrutkar, Patrick Traynor, and Paul van Oorschot. An Empirical Evaluation of Security Indicators in Mobile Web Browsers. *IEEE Transactions on Mobile Computing (TMC)*, 14(5), 2015.

15. Andrew Harris, Seymour Goodman, and Patrick Traynor. Privacy and Security Concerns Associated with Mobile Money Applications in Africa. *Washington Journal of Law, Technology & Arts*, 8(3), 2013.

16. Italo Dacosta, Saurabh Chakradeo, Mustaque Ahamad, and Patrick Traynor. One-Time Cookies: Preventing Session Hijacking Attacks with Stateless Authentication Tokens. *ACM Transactions on Internet Technology (TOIT)*, 12(1), 2012.

17. Cong Shi, Xiapu Luo, Patrick Traynor, Mostafa Ammar, and Ellen Zegura. ARDEN: Anonymous netwoRking in Delay tolErant Networks. *Journal of Ad Hoc Networks*, 10(6):918–930, 2012.

18. Patrick Traynor. Characterizing the Security Implications of Third-Party EAS Over Cellular Text Messaging Services. *IEEE Transactions on Mobile Computing (TMC)*, 11(6):983–994, 2012.

19. Italo Dacosta, Vijay Balasubramaniyan, Mustaque Ahamad, and Patrick Traynor. Improving Authentication Performance of Distributed SIP Proxies. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 22(11):1804–1812, 2011.

20. Patrick Traynor, Chaitrali Amrutkar, Vikhyath Rao, Trent Jaeger, Patrick McDaniel, and Thomas La Porta. From Mobile Phones to Responsible Devices. *Journal of Security and Communication Networks (SCN)*, 4(6):719 – 726, 2011.

21. Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure Attribute-Based Systems. *Journal of Computer Security (JCS)*, 18(5):799–837, 2010.

22. Patrick Traynor, Kevin Butler, William Enck, Kevin Borders, and Patrick McDaniel. malnets: Large-Scale Malicious Networks via Compromised Wireless Access Points. *Journal of Security and Communication Networks (SCN)*, 2(3):102–113, 2010.

23. Patrick Traynor. Securing Cellular Infrastructure: Challenges and Opportunities. *IEEE Security & Privacy Magazine*, 7(4), 2009.

24. Kevin Butler, Sunam Ryu, Patrick Traynor, and Patrick McDaniel. Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 20(12):1803–1815, 2009.

25. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating Attacks On Open Functionality in SMS-Capable Cellular Networks. *IEEE/ACM Transactions on Networking (TON)*, 17(1), 2009.

26. Patrick Traynor, Michael Chien, Scott Weaver, Boniface Hicks, and Patrick McDaniel. Non-Invasive Methods for Host Certification. *ACM Transactions on Information and System Security (TISSEC)*, 2008.

27. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. *Journal of Computer Security (JCS)*, 16(6):713–742, 2008.

28. Patrick Traynor, Raju Kumar, Heesook Choi, Sencun Zhu, Guohong Cao, and Thomas La Porta. Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. *IEEE Transactions on Mobile Computing (TMC)*, 6(6), 2007.

C. Published Books and Parts of Books

1. Andrew Harris, Frank S. Park, Seymour Goodman, and Patrick Traynor. *Emerging Privacy and Security Concerns for Digital Wallet Deployment*. Privacy in America: Interdisciplinary Perspectives. Scarecrow Press, July 2011.

2. Kevin Butler, William Enck, Patrick Traynor, Jennifer Plasterr, and Patrick McDaniel. *Privacy Preserving Web-Based Email*. Algorithms, Architectures and Information Systems Security, Statistical Science and Interdisciplinary Research. World Scientific Computing, November 2008.

3. Patrick Traynor, Patrick McDaniel, and Thomas La Porta. *Security for Telecommunications Networks*. Number 978-0-387-72441-6 in Advances in Information Security Series. Springer, August 2008.

D. Edited Proceedings

None.

E. Conference Presentations

E.1. Conference Papers with Proceedings (Refereed)

1. Nathaniel Bennett, Tyler Tucker, Carson Stillman, William Enck, Patrick Traynor, and Kevin Butler. Fizzle: A Framework for Deterministic and Reproducible Network Fuzzing. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2026.

2. Seth Layton, Bernardo Medeiros, Kevin Butler, and Patrick Traynor. AI Wrote My Paper and All I Got Was This False Negative: Measuring the Efficacy of Commercial AI Text Detectors. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2026.

3. Carson Stillman, Jennifer Sheldon, Ian Garrett, Patrick Traynor, Ryan Gerdes, Sara Rampazzi, and Kevin Butler. EM-Flow: Advanced Electromagnetic Control Flow Verification for Embedded Systems. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2025.

4. Daniel Olszewski, Tyler Tucker, Kevin Butler, and Patrick Traynor. SoK: Towards a Unified Approach to Applied Replicability for Computer Security. In *Proceedings of the USENIX Security Symposium (Security)*, 2025.

5. Daniel Olszewski, Allison Lu, Anna Crowder, Nathaniel Bennett, Seth Layton, Sri Hrushikesh Varma Bhupathiraju, Tyler Tucker, Siddhant Kalgutkar, Hunter Ver Helst, Carson Stillman, Kevin Butler, Sara Rampazzi, and Patrick Traynor. "Raise Your Hand If You've Been Personally Victimized By A Lack Of Reproducibility": On Reproducibility in Tier 2 Security Conferences. In *Proceedings of ACM Conference on Reproducibility and Replicability (REP)*, 2025.

6. Cassidy Gibson and Daniel Olszewski and Natalie Grace Brigham and Anna Crowder and Kevin R. B. Butler and Patrick Traynor and Elissa M. Redmiles and Tadayoshi Kohno. Analyzing the AI Nudification Application Ecosystem. In *Proceedings of the USENIX Security Symposium (Security)*, 2025.

7. Tyler Tucker, Nathaniel Bennett, Martin Kotuliak, Simon Erni, Srdjan Capkun, Kevin Butler, and Patrick Traynor. Detecting IMSI-Catchers by Characterizing Identity Exposing Messages in Cellular Traffic. In *Symposium on Network and Distributed System Security (NDSS)*, 2025. (Acceptance Rate 16.1%).

8. Anna Crowder, Allison Lu, Kevin Childs, Carson Stillman, Patrick Traynor, and Kevin Butler. Data to Infinity and Beyond: Examining Data Sharing and Reuse Practices in the Computer Security Community. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2025.

9. Magdalena Pasternak, Kevin Warren, Daniel Olszewski, Susan Nittroueri, Patrick Traynor, and Kevin Butler. Characterizing the Impact of Audio Deepfakes in the Presence of Cochlear Implant Simulated Audio. In *Symposium on Network and Distributed System Security (NDSS)*, 2025. (Acceptance Rate 16.1%).

10. Anna Crowder, Daniel Olszewski, Patrick Traynor, and Kevin R. B. Butler. I Can Show You the World (of Censorship): Extracting Insights from Censorship Measurement Data Using Statistical Techniques. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, December 2024. (Acceptance Rate 21.8%).

11. Kevin Childs, Cassidy Gibson, Anna Crowder, Kevin Warren, Carson Stillman, Elissa Redmiles, Eakta Jain, Patrick Traynor, and Kevin Butler. "I Had Sort of a Sense that I Was Always Being Watched... Since I Was": Examining Interpersonal Discomfort From Continuous Location-Sharing Applications. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2024. (Acceptance Rate 16.9%).

12. Nathaniel Bennett, Weidong Zhu, Benjamin Simon, Ryon Kennedy, William Enck, Patrick Traynor, and Kevin Butler. RANsacked: A Domain-Informed Approach for Fuzzing LTE and 5G RAN-Core Interfaces. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2024. (Acceptance Rate 16.9%).

13. Kevin Warren, Tyler Tucker, Anna Crowder, Daniel Olszewski, Allison Lu, Caroline Fedele, Magdalena Pasternak, Seth Layton, Kevin Butler, Carrie Gates, and Patrick Traynor. Better Be Computer or I'm Dumb": A Large-Scale Evaluation of Humans as Audio Deepfake Detectors. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2024. (Acceptance Rate 16.9%).

14. K. Virgil English, Nathaniel Bennett, Seaver Thorn, Kevin Butler, William Enck, and Patrick Traynor. Examining Cryptography and Randomness Failures in Open-Source Cellular Cores. In *Proceedings of the ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2024. (Acceptance rate: 21.3%)(Best Paper).

15. Seth Layton, Tyler Tucker, Daniel Olszewski, Kevin Warren, Carrie Gates, Kevin Butler, and Patrick Traynor. SoK: The Good, The Bad, and The Unbalanced: Measuring Structural Limitations of Current Deepfake Datasets. In *Proceedings of the USENIX Security Symposium (Security)*, 2024. (Acceptance Rate 18.3%).

16. Imani Munyaka, Daniel Delgado, Juan Gilbert, Jaime Ruiz, and Patrick Traynor. "I used to live in Florida": Exploring the Impact of Spam Call Warning Accuracy on Callee Decision-Making. In *Symposium on Usable Security and Privacy (USEC)*, 2024.

17. Jianliang Wu, Patrick Traynor, Dongyan Xu, Dave (Jing) Tian, and Antonio Bianchi. Finding Traceability Attacks in the Bluetooth Low Energy Specification and Its Implementations. In *Proceedings of the USENIX Security Symposium (Security)*, 2024. (Acceptance Rate 18.3%).

18. Daniel Olszewski, Allison Lu, Carson Stillman, Kevin Warren, Cole Kitroser, Alejandro Pascual, Divyajyoti Ukirde, Kevin Butler, and Patrick Traynor. "Get in Researchers; We're Measuring Reproducibility": A Reproducibility Study of Machine Learning Papers in Tier 1 Security Conferences. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2023. (Acceptance rate: 19.8%).

19. Christian Peeters, Tyler Tucker, Anushri Jain, Kevin Butler, and Patrick Traynor. LeopardSeal: Detecting Call Interception via Audio Rogue Base Stations. In *Proceedings of the ACM International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2023. (Acceptance rate: 21%).

20. Tyler Tucker, Hunter Searle, Kevin Butler, and Patrick Traynor. Blue's Clues: Practical Discovery of Non-Discoverable Bluetooth Devices. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2023. (Acceptance rate: 14%).

21. Hadi Abdullah, Aditya Karlekar, Saurabh Prasad, Muhammad Sajidur Rahman, Logan Blue, Luke Bauer, Vincent Bindschaedler, and Patrick Traynor. Attacks as Defenses: Designing Robust Audio CAPTCHAs Using Attacks on Automatic Speech Recognition Systems. In *Symposium on Network and Distributed System Security (NDSS)*, 2023. (Acceptance rate: 16%).

22. Daniel Olszewski, Sandeep Sathyanarayana, Weidong Zhu, Kevin Butler, and Patrick Traynor. HallMonitor: A Framework for Identifying Network Policy Violations in Software. In *IEEE Conference on Communications and Network Security (CNS)*, 2022.

23. Hadi Abdullah, Aditya Karlekar, Vincent Bindschaedler, and Patrick Traynor. Demystifying Limited Adversarial Transferability in Automatic Speech Recognition Systems. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. (Acceptance rate: 32%).

24. Logan Blue, Kevin Warren, Hadi Abdullah, Cassidy Gibson, Luis Vargas, Jessica O'Dell, Kevin Butler, and Patrick Traynor. Who Are You (I Really Wanna Know)? Detecting Audio DeepFakes Through Vocal Tract Reconstruction. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2022. (Acceptance rate: 17.2%).

25. Grant Hernandez, Marius Muench, Dominik Maier, Alyssa Milburn, Shinjo Park, Tobias Scharnowski, Tyler Tucker, Patrick Traynor, and Kevin R. B. Butler. FirmWire: Transparent Dynamic Analysis for Cellular Baseband Firmware. In *Symposium on Network and Distributed System Security (NDSS)* , 2022. (Acceptance rate: 16.2%).

26. Christian Peeters, Christopher Patton, Imani N. Sherman, Daniel Olszewski, Thomas Shrimpton, and Patrick Traynor. SMS OTP Security (SOS): Hardening SMS-Based Two Factor Authentication. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2022. (Acceptance rate: 18.2%).

27. Hadi Abdullah, Muhammad Sajidur Rahman, Christian Peeters, Cassidy Gibson, Washington Garcia, Vincent Bindschaedler, Thomas Shrimpton, and Patrick Traynor. Beyond L_p Clipping: Equalization based Psychoacoustic Attacks against ASRs. In *The Asian Conference on Machine Learning (ACML)*, 2021.

28. Imani Sherman and Daniel Delgado and Juan Gilbert and Jaime Ruiz and Patrick Traynor. Characterizing User Comprehension in the STIR/SHAKEN Anti-Robocall Standard. In *Proceedings of the Annual Research Conference on Comunications Information and Internet Policy (TPRC 49)*, 2021.

29. Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021. (Acceptance rate: 12.1%).

30. Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Logan Blue, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. Hear "No Evil", See "Kenansville": Efficient and Transferable Black-Box Attacks on Speech Recognition and Voice Identification Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021. (Acceptance rate: 12.1%).

31. Imani Sherman, Jasmine Bowers, Liz-Laure Laborde, Juan E. Gilbert, Jaime Ruiz, and Patrick Traynor. Truly Visual Caller ID? An Analysis of Anti-Robocall Applications and their Accessibility to Visually Impaired Users. In *IEEE International Symposium on Technology and Society (IEEE ISTAS)*, 2020.

32. Imani Sherman, Jasmine Bowers, Keith McNamara, Juan Gilbert, Jaime Ruiz, and Patrick Traynor. Are You Going to Answer That? Measuring User Responses to Anti-Robocall Application Indicators. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2020. (Acceptance rate: 17.4%).

33. Joseph Choi, Dave Tian, Grant Hernandez, Christopher Patton, Benjamin Mood, Thomas Shrimpton, Patrick Traynor, and Kevin Butler. A Hybrid Approach to Secure Function Evaluation Using SGX. In *Procedings of the ACM ASIA Conference on Computer and Communications Security (ASIACCS'19)*, 2019. (Acceptance Rate: 17.0% for full papers).

34. Vanessa Frost, Dave Tian, Christie Ruales, Patrick Traynor, and Kevin Butler. Examining DES-based Cipher Suite Support within the TLS Ecosystem. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACCS'19)*, 2019. (Acceptance Rate: 22.0% for all papers).

35. Dave Tian, Joseph Choi, Grant Hernandez, Patrick Traynor, and Kevin Butler. A Practical Intel SGX Setting for Linux Containers in the Cloud. In *Proceedings of the ACM Conference on Data and Application Security and Privacy (CODASPY'19)*, 2019. (Acceptance rate: 23.5%).

36. Nolen Scaife, Jasmine Bowers, Christian Peeters, Grant Hernandez, Imani Sherman, Lisa Anthony, and Patrick Traynor. Kiss from a Rogue: Evaluating Detectability of Pay-at-the-Pump Card Skimmers. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2019. (Acceptance rate: 12.0%).

37. Jasmine Bowers, Imani Sherman, Kevin Butler, and Patrick Traynor. Characterizing Security and Privacy Practices in Emerging Digital Credit Applications. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019. (Acceptance rate: 25.6%).

38. Hadi Abdullah, Washington Garcia, Christian Peeters, P. Traynor, K. Butler, and J. Wilson. Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2019. (Acceptance Rate: 17.1%).

39. Lius Vargas, Logan Blue, Vanessa Frost, Christopher Patton, N. Scaife, K. Butler, and P. Traynor. Digital Healthcare-Associated Infection Analysis of a Major Multi-Campus Hospital System. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2019. (Acceptance Rate: 17.1%).

40. Dominik Wermke, Nicolas Huaman, Yasemin Acar, Bradley Reaves, Patrick Traynor, and Sascha Fahl. A Large Scale Investigation of Obfuscation Use in Google Play. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2018. Acceptance Rate: 20.1%.

41. Nolen Scaife, Christian Peeters, and Patrick Traynor. Fear the Reaper: Characterization and Fast Detection of Card Skimmers. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2018. Acceptance Rate: 19.0%.

42. Dave (Jing) Tian, Grant Hernandez, Joseph Choi, Vanessa Frost, Christie Raules, Kevin Butler, Patrick Traynor, Hayawardh Vijayakumar, Lee Harrison, Amir Rahmati, and Mike Grace. ATtention Spanned: Comprehensive Vulnerability Analysis of AT Commands Within the Android Ecosystem. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2018. Acceptance Rate: 19.0%.

43. Luis Vargas, Gyan Hazarika, Rachel Culpepper, Kevin Butler, Thomas Shrimpton, Doug Szajda, and Patrick Traynor. Mitigating Risk while Complying with Data Retention Laws. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2018.

44. Logan Blue, Luis Vargas, and Patrick Traynor. Hello, Is It Me You're Looking For? Differentiating Between Human and Electronic Speakers for Voice Interface Security. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2018.

45. Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2MA: Verifying Voice Commands via Two Microphone Authentication. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2018. (Acceptance Rate: 20.0%).

46. Nolen Scaife, Christian Peeters, Camilo Velez, Hanqing Zhao, Patrick Traynor, and David Arnold. The Cards Aren't Alright: Detecting Counterfeit Gift Cards Using Encoding Jitter. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2018. (Acceptance Rate: 10.4%).

47. Christian Peeters, Hadi Abdullah, Nolen Scaife, Jasmine Bowers, Patrick Traynor, Bradley Reaves, and Kevin Butler. Sonar: Detecting SS7 Redirection Attacks Via Call Audio-Based Distance Bounding. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2018. (Acceptance Rate: 10.4%).

48. Tyler Ward, Joseph Choi, Kevin Butler, John M. Shea, Patrick Traynor, and Tan Wong. Privacy Preserving Localization Using a Distributed Particle Filtering Protocol. In *IEEE MILCOM*, 2017. (Acceptance Rate: 56%).

49. Bradley Reaves and Logan Blue and Hadi Abdullah and Luis Vargas and Patrick Traynor and Thomas Shrimpton. AuthentiCall: Efficient Identity and Content Authentication for Phone Calls. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2017. (Acceptance Rate: 16.3%).

50. Jasmine Bowers and Bradley Reaves and Imani N. Sherman and Patrick Traynor and Kevin Butler. Regulators, Mount Up! Analysis of Privacy Policies for Mobile Money Applications. In *Proceedings of the USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2017. (Acceptance Rate: 26.5%).

51. Bradley Reaves, Logan Blue, and Patrick Traynor. AuthLoop: End-to-End Cryptographic Authentication for Telephony over Voice Channels. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2016. (Acceptance Rate: 15.5%).

52. Dave Tian, Nolen Scaife, Adam Bates, Kevin Butler, and Patrick Traynor. Making USB Great Again with USBFILTER. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2016. (Acceptance Rate: 15.5%).

53. Bradley Reaves, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Detecting SMS Spam in the Age of Legitimate Bulk Messaging. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2016. (Acceptance Rate: 35.0%).

54. Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin Butler. CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2016. (Acceptance Rate: 17.6%).

55. Bradley Reaves, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Sending out an SMS: Characterizing the Security of the SMS Ecosystem with Public Gateways. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2016. (Acceptance Rate: 13.0%).

56. Benjamin Mood, Debayan Gupta, Henry Carter, Kevin Butler, and Patrick Traynor. Frigate: A Validated, Extensible, and Efficient Compiler and Interpreter for Secure Computation. In *Proceedings of the IEEE European Symposium on Security and Privacy*, 2016. (Acceptance Rate: 17.3%).

57. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Outsourcing Secure Two-Party Computation as a Black Box. In *Proceedings of the International Conference on Cryptology and Network Security*, 2015. (Acceptance Rate: 52.9%).

58. Nolen Sciafe, Henry Carter, and Patrick Traynor. OnionDNS: A Seizure-Resistant Top-level Domain. In *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, 2015. (Acceptance Rate: 28.1%).

59. Bradley Reaves, Nolen Scaife, Adam Bates, Patrick Traynor, and Kevin Butler. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2015. (Acceptance Rate: 15.7%).

60. Bradley Reaves, Ethan Shernan, Adam Bates, Henry Carter, and Patrick Traynor. Boxed Out: Blocking Cellular Interconnect Bypass Fraud at the Network Edge. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2015. (Acceptance Rate: 15.7%).

61. David Dewey, Bradley Reaves, and Patrick Traynor. Uncovering Use-After-Free Conditions In Compiled Code. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES)*, 2015. (Acceptance Rate: 22%).

62. Ethan Shernan, Henry Carter, Dave Tian, Patrick Traynor, and Kevin Butler. More Guidelines Than Rules: CSRF Vulnerabilities from Noncompliant OAuth 2.0 Implementations. In *Proceedings of the International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA)*, 2015. (Acceptance Rate: 22.7%).

63. Henry Carter, Charles Lever, and Patrick Traynor. Whitewash: Outsourcing Garbled Circuit Generation for Mobile Devices. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2014. (Acceptance Rate: 19.9%).

64. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Secure Outsourced Garbled Circuit Evaluation for Mobile Devices. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2013. (Acceptance Rate: 16.2%).

65. Chaitrali Amrutkar, Matti Hiltunen, Shobha Venkataraman, Kaustubh Joshi, Patrick Traynor, Trevor Jim, and Oliver Spatscheck. Why is My Smartphone Slow? On The Fly Diagnosis of Poor Performance on the Mobile Internet. In *Proceedings of The 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2013. (Acceptance Rate: 19.6%).

66. Saurabh Chakradeo, Brad Reaves, Patrick Traynor, and William Enck. MAST: Triage for Market-scale Mobile Malware Analysis. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2013. (Acceptance Rate: 15.0%)(Best Paper).

67. Charles Lever, Manos Antonakakis, Brad Reaves, Patrick Traynor, and Wenke Lee. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2013. (Acceptance rate: 18.8%).

68. Chaitrali Amrutkar, Kapil Singh, Arunabh Verma, and Patrick Traynor. VulnerableMe: Measuring Systemic Weaknesses in Mobile Browser Security. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, 2012. (Acceptance rate: 25%) (Best Paper - SAIC Student Paper Competition (GT)) (Finalist - CSAW AT&T Applied Security Research Best Paper Competition 2012).

69. Chaitrali Amrutkar, Patrick Traynor, and Paul van Oorschot. A Measurement Study of SSL Indicators on Mobile Browsers: Extended Life, or End of the Road? In *Proceedings of the Information Security Conference (ISC)*, 2012. (Acceptance rate: 32%) (Best Student Paper).

70. Italo Dacosta, Mustaque Ahamad, and Patrick Traynor. Trust No One Else: Detecting MITM Attacks Against SSL/TLS Without Third-Parties. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, 2012. (Acceptance Rate: 20.2%).

71. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. Accountable Wiretapping -or- I Know They Can Hear You Now. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2012. (Acceptance Rate: 17.8%).

72. Yacin Nadji, Jon Giffin, and Patrick Traynor. Automated Remote Repair for Mobile Malware. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011. (Acceptance Rate: 18.5%).

73. Nilesh Nipane, Italo Dacosta, and Patrick Traynor. "Mix-In-Place" Anonymous Networking Using Secure Function Evaluation. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011. (Acceptance Rate: 18.5%).

74. Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2011. (Acceptance Rate: 13.9%).

75. Philip Marquardt, David Dagon, and Patrick Traynor. Impeding Individual User Profiling in Shopper Loyalty Programs. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*, 2011. (Acceptance Rate: 35.1%).

76. David Dewey and Patrick Traynor. No Loitering: Exploiting Lingering Vulnerabilities in Default COM Objects. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2011. (Acceptance Rate: 20.1%).

77. Vijay Balasubramaniyan, Aamir Poonawalla, Mustaque Ahamad, Michael Hunter, and Patrick Traynor. PinDr0p: Using Single-Ended Audio Features to Determine Call Provenance. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2010. (Acceptance Rate: 17.2%).

78. Patrick Traynor, Joshua Schiffman, Thomas La Porta, Patrick McDaniel, Abhrajit Ghosh, and Farooq Anjum. Constructing Secure Localization Systems with Adjustable Granularity. In *IEEE Global Communications Conference (GLOBECOM)*, 2010. (Acceptance Rate: 35.6%).

79. Patrick Traynor. Characterizing the Security Implications of Third-Party EAS Over Cellular Text Messaging Services. In *Proceedings of the Second IEEE International Conference on Security and Privacy in Communication Networks (SecureComm)*, 2010. (Acceptance Rate: 25.0%).

80. Kapil Singh, Samrit Sangal, Nehil Jain, Patrick Traynor, and Wenke Lee. Evaluating Bluetooth as a Medium for Botnet Command and Control. In *Proceedings of the International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA)*, 2010. (Acceptance Rate: 30.7%).

81. Italo Dacosta and Patrick Traynor. Proxychain: Developing a Robust and Efficient Authentication Infrastructure for Carrier-Scale VoIP Networks. In *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2010. (Acceptance Rate: 17.0%).

82. Frank S. Park, Chinmay Gangakhedkar, and Patrick Traynor. Leveraging Cellular Infrastructure to Improve Fraud Prevention. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2009. (Acceptance Rate: 19.0%).

83. Patrick Traynor, Michael Lin, Machigar Ongtang, Vikyath Rao, Trent Jaeger, Thomas La Porta, and Patrick McDaniel. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2009. (Acceptance Rate: 18.4%).

84. Brendan Dolan-Gavitt, Abhinav Srivastava, Patrick Traynor, and Jonathon Giffin. Robust Signatures for Kernel Data Structures. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2009. (Acceptance Rate: 18.4%).

85. Italo Dacosta, Vijay Balasubramaniyan, Mustaque Ahamad, and Patrick Traynor. Improving Authentication Performance of Distributed SIP Proxies. In *Proceedings of the Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm)*, 2009. (Acceptance Rate: 43.3%).

86. Patrick Traynor, Kevin Butler, William Enck, and Patrick McDaniel. Realizing Massive-Scale Conditional Access Systems Through Attribute-Based Cryptosystems. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2008. (Acceptance Rate: 17.7%).

87. Patrick Traynor, Patrick McDaniel, and Thomas La Porta. On Attack Causality in Internet-Connected Cellular Networks. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2007. (Acceptance Rate: 12.3%).

88. Sunam Ryu, Kevin Butler, Patrick Traynor, and Patrick McDaniel. Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems. In *Proceedings of the IEEE International Symposium on Security in Networks and Distributed Systems (SSNDS)*, 2007. (Acceptance Rate: 40%).

89. Luke St. Clair, Lisa Johansen, William Enck, Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Trent Jaeger. Password Exhaustion: Predicting the End of Password Usefulness. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, 2006. (Invited Paper).

90. Kevin Butler, William Enck, Jennifer Plasterr, Patrick Traynor, and P. McDaniel. Privacy-Preserving Web-Based Email. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, December 2006. (Acceptance Rate: 30.4%).

91. Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure Attribute-Based Systems. In *Proceedings of the Thirteenth ACM Conference on Computer and Communications Security (CCS)*, November 2006. (Acceptance Rate: 14.8%).

92. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the Twelfth Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2006. (Acceptance Rate: 11.7%).

93. Patrick Traynor, Michael Chien, Scott Weaver, Boniface Hicks, and Patrick McDaniel. Non-Invasive Methods for Host Certification. In *Proceedings of the Second IEEE International Conference on Security and Privacy in Communication Networks (SecureComm)*, August 2006. (Acceptance Rate: 25.4%).

94. Patrick Traynor, JaeShung Shin, Barat Madan, Shashi Phoha, and Thomas La Porta. Efficient Group Mobility for Heterogeneous Sensor Networks. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall)*, September 2006. (Acceptance Rate: 58%).

95. Patrick Traynor, Raju Kumar, Hussain Bin Saad, Guohong Cao, and Thomas La Porta. LIGER: Implementing Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. In *Proceedings of the 4th ACM International Conference on Mobile Systems, Applications and Services (MobiSys)*, June 2006. (Acceptance Rate: 15.4%).

96. Patrick Traynor, Guohong Cao, and Thomas La Porta. The Effects of Probabilistic Key Management on Secure Routing in Sensor Networks. In *Proceedings of the 2006 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2006. (Acceptance Rate: 38.8%).

97. Patrick Traynor, Heesook Choi, Guohong Cao, Sencun Zhu, and Thomas La Porta. Establishing Pair-Wise Keys In Heterogeneous Sensor Networks. In *Proceedings of the 25th Annual IEEE Conference on Computer Communications (INFOCOM)*, April 2006. (Acceptance Rate: 18%).

98. William Enck, Patrick Traynor, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the Twelfth ACM Conference on Computer and Communications Security (CCS)*, November 2005. (Acceptance Rate: 15%).

Removed for external version.

E.2. Conference Presentations with Proceedings (Non-Refereed)

None.

E.3. Conference Presentations without Proceedings

1. Patrick Traynor. Work in Progress Presentations: Fine-Grained Secure Localization for 802.11 Networks. 15th USENIX Security Symposium (SECURITY), August 2006.
2. Patrick Traynor. Work in Progress Presentations: Fundamental Limitations of Sensor Network Security. ACM/USENIX Fourth International Conference on Mobile Systems Applications and Services (MobiSys), June 2006. (Award: Most Entertaining WIP).
3. Patrick Traynor, Heesook Choi, Guohong Cao, and Thomas La Porta. Poster Session: Probabilistic Unbalanced Key Distribution and Its Effects on Distributed Sensor Networks. Workshop on Wireless Security (WiSe), October 2004.

F. Other

F.1. Submitted Journal Papers

None.

F.2. Refereed Research Reports

None.

F.3. Software

1. *GSM Air Interface Simulator:* Developed a full voice, data and SMS capable simulator for the wireless portion of a GSM network. Models communications down to the timeslot for highest possible accuracy. Used in the majority of our work on cellular security.
2. *Malicious Telephony Load Tester:* Built a system on top of the TM1 Telecom Database testing suite to allow for a comparison of malicious traffic of varying composition.

F.4. Published Papers (Non-Refereed)

1. Patrick Traynor. Characterizing the Limitations of Third-Party EAS Over Cellular Text Messaging Services. Technical report, 3G Americas Whitepaper, 2008.
2. Lisa Johansen, Kevin Butler, William Enck, Patrick Traynor, and Patrick McDaniel. Grains of SANs: Building Storage Area Networks from Memory Spots. Technical Report NAS-TR-0060-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, 2007.

F.5. Books in Preparation

None.

F.6. Workshops and External Courses

1. Luis Vargas, Patrick Emami, and Patrick Traynor. On the Detection of Disinformation Campaign Activity with Network Analysis. In *Proceedings of the 2020 ACM SIGSAC Cloud Computing Security Workshop*, CCSW '20, 2020.

2. Siddhant Deshmukh, Henry Carter, Grant Hernandez, Patrick Traynor, and Kevin Butler. Efficient and Secure Template Blinding for Biometric Authentication. In *IEEE Workshop on Security and Privacy in the Cloud (SPC)*, 2016.

3. Debayan Gupta, Benjamin Mood, Joan Feigenbaum, Kevin Butler, and Patrick Traynor. Using Intel Software Guard Extensions for Efficient Two-Party Secure Function Evaluation. In *Workshop on Encrypted Computing and Applied Homomorphic Cryptography (WAHC)*, 2016.

4. Chaitrali Amrutkar and Patrick Traynor. Rethinking Permissions for Mobile Web Apps: Barriers and the Road Ahead. In *Proceedings of the ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*, 2012.

5. Nigel Lawrence and Patrick Traynor. Under New Management: Practical Attacks on SNMPv3. In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*, 2012.

6. Andrew Harris, Frank S. Park, Seymour Goodman, and Patrick Traynor. Emerging Privacy Concerns for Digital Wallet Deployment. In *Proceedings of the Workshop on Making Privacy in America*, 2009.

7. Patrick Traynor. Privacy and Security Concerns for Personal and Mobile Health Devices. In *Proceedings of the Workshop to Set A Research Agenda for Privacy and Security of Healthcare Technologies*, 2009.

8. Kevin Butler, William Enck, Harri Hursti, Stephen McLaughlin, Patrick Traynor, and Patrick McDaniel. Systemic Issues in the Hart InterCivic and Premier Voting System: Reflections Following Project EVEREST. In *Proceedings of the USENIX/ACCURATE Electronic Voting Technology (EVT) Workshop*, 2008.

G. Research Proposals and Grants (Principal Investigator)

1. Approved and Funded

1. **Artus Protocol STTR Phase II - Extension**
   Sponsor: Office of Naval Research
   Investigator(s): Patrick Traynor (PI), Kevin Butler
   Amount: *$300,000 over 2 years*
   Awarded: August 2023

2. **Testing Audio Deep Fake Detectors**
   Sponsor: Bank of America
   Investigator(s): Patrick Traynor (PI), Kevin Butler
   Amount: *$150,000 over 1 year*
   Awarded: August 2023

3. **Testing Audio Deep Fake Detectors**
   Sponsor: Bank of America
   Investigator(s): Patrick Traynor (PI), Kevin Butler
   Amount: *$274,000 over 2 years*
   Awarded: August 2021

4. **Deploying Defenses for Cellular Networks Using the AWARE Testbed**
   Sponsor: Department of Homeland Security: CISA:
   Investigator(s): Patrick Traynor (PI), Kevin Butler, Guofei Gu, Radu Stoleru, Walter Magnussen, P. R. Kumar
   Amount: *$3,100,000 over 4 years*
   Awarded: October 2019

5. **SaTC:CORE:Medium: Securing the Voice Processing Pipeline Against Adversarial Audio**
   Sponsor: NSF Secure and Trustworthy Cyberspace
   Investigator(s): Patrick Traynor (PI), Thomas Shrimpton, Vincent Bindschaedler
   Amount: *$1,199,999 over 4 years*
   Awarded: October 2019

6. **Artus Protocol STTR Phase II**
   Sponsor: Office of Naval Research
   Investigator(s): Patrick Traynor (PI), Kevin Butler
   Amount: *$800,000 over 4 years*
   Awarded: August 2019

7. **Evaluating the Security of QR Code-Based Payments**
   Sponsor: Discover Financial
   Investigator(s): Patrick Traynor (PI)
   Amount: *$50,000 over 1 year*
   Awarded: September 2018

8. **Workshop: Addressing the Technical Security Challenges of Emerging Digital Financal Services**
   Sponsor: NSF Secure and Trustworthy Cyberspace
   Investigator(s): Patrick Traynor (PI), Kevin Butler
   Amount: *$50,000 over 1 year*
   Awarded: September 2017

9. **Designing Strong End-to-End Authentication Mechanisms for Modern Telephony Systems**
   Sponsor: NSF Secure and Trustworthy Cyberspace
   Investigator(s): Patrick Traynor (PI)
   Amount: *$500,000 over 3 years*
   Awarded: July 2016

10. **Digital Healthcare-Associated Infection: Measurement, Defense and Prevention in a Modern Digital Healthcare Ecosystem**
    Sponsor: National Science Foundation
    Investigator(s): Patrick Traynor (PI), Kevin Butler, Shigang Chen
    Amount: *$1,200,000 over 4 years*
    Awarded: June 2016

11. **Evaluating and Improving Security in Emerging Branchless Banking Systems**
    Sponsor: NSF Secure and Trustworthy Cyberspace
    Investigator(s): Patrick Traynor (PI)
    Amount: *$500,000 over 3 years*
    Awarded July 2015

12. **Prevention and Detection of Disallowed Connections in Mobile and Pervasive Systems**
    Sponsor: CISE-ECE Harris Endowed Seed Fund Program
    Investigator(s): Patrick Traynor (PI), Renato Figueiredo (PI)
    Amount: *$40,000 over 1 year*
    Awarded December 2014

13. **Mobile Excursion Study Support**
Sponsor: Hanscom AFB Electronic Systems Command Development Planning Division (ESC/XR)
Investigator(s): Patrick Traynor (PI), Mustaque Ahamad, Jeff Evans, Chuck Bokath
Amount: *$280,000 over 3 months*
Awarded July 2012

14. **Characterizing the Security Limitations of Accessing the Mobile Web**
Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI) and William Enck (NC State)
Amount: *$334,000 over 3 years*
Awarded July 2012

15. **Mitigating Attacks on Mobile Devices and Critical Cellular Infrastructure**
Sponsor: US Department of Defense - Defense University Research Instrumentation Program (DURIP)
Investigator(s): Patrick Traynor (PI), Jon Giffin, Mustaque Ahamad
Amount: *$210,081 over 1 year*
Awarded June 2011

16. **Characterizing and Implementing Efficient Primitives for Privacy-Preserving Computation**
Sponsor: DARPA PROgramming Computation on EncryptEd Data (PROCEED) – Broad Agency Announcement
Investigator(s): Patrick Traynor (PI) and Kevin Butler (UOregon)
Amount: *$580,000 over 4 years*
Awarded May 2011

17. **Security for Converged IMS Networks**
Sponsor: US Department of Defense
Investigator(s): Patrick Traynor (PI), Mustaque Ahamad and Russ Clark
Amount: *$242,401 over 1 year*
Awarded August 2010

18. **CAREER: Protecting User Data on Lost, Stolen and Damaged Mobile Phones**
Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI)
Amount: *$400,000 over 5 years*
Awarded: May 2010

19. **Provably Anonymous Networking Through Secure Function Evaluation**
Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI)
Amount: *$200,000 over 2 years*
Awarded: July 2009

20. **Characterizing and Mitigating Device-Based Attacks in Cellular Telecommunications Networks**
Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI) and Jonathon Giffin
Amount: *$450,000 over 3 years*
Awarded: July 2009

2. Pending

Removed for external version.

H. Research Proposals and Grants (Contributor)

1. Approved and Funded

1. **SaTC: Frontier: Securing the Future of Computing for Marginalized and Vulnerable Populations**
   Sponsor: NSF SaTC
   Investigator(s): Kevin Butler (PI), Patrick Traynor, Tadayoshi Kohno, Franzi Roesner, Apu Kapadia, Eakta Jain.
   Amount: *$7,500,000 for 5 years*
   Awarded October 2022

2. **ROCKY: Reliable Obfuscated Communications Kit for everYone**
   Sponsor: DARPA Resilient Anonymous Communication for Everyone (RACE) – Broad Agency Announcement
   Investigator(s): Thomas Shrimpton (PI), Patrick Traynor, Kevin Butler, Vincent Bindschaedler, Nadia Heninger
   Amount: *$1,600,000 over 4 years*
   Awarded May 2019

3. **WiFiUS: Collaborative Research: SELIOT: Securing Lifecycle of Internet-of-Things**
   Sponsor: NSF CNS WiFiUS
   Investigator(s): Gene Tsudik (PI), Patrick Traynor
   Amount: *$300,000 for 2 years*
   Submitted December 2016

4. **Cloud-based Oblivious Spectrum Mapping and Allocation**
   Sponsor: NSF CNS EARS
   Investigator(s): John Shea (PI), Tan Wong, Patrick Traynor
   Amount: *$532,952 for 2 years*
   Submitted May 2016

5. **DURIP: Developing Research Capability in Cyber-Physical Systems at the University of Florida**
   Sponsor: Small
   Investigator(s): Kevin Butler (PI), Patrick Traynor, My Thai
   Amount: *$200,000 for 2 years*
   Submitted: June 2015

6. **Securing the New Converged Telephony Landscape**
   Sponsor: NSF TWC: Small
   Investigator(s): Mustaque Ahamad (PI) and Patrick Traynor
   Amount: *$500,000 for 3 years*
   Submitted: December 2012

7. **Facilitating Free and Open Access to Information on the Internet**
   Sponsor: NSF Trustworthy Computing
   Investigator(s): Nick Feamster (PI), Wenke Lee, Patrick Traynor, Hans Klein, Roger Dingledine, Michael Freedman and Edward W. Felten
   Amount: *$1,500,000 for 4 years*
   Awarded: June 2011

8. **Monitoring Free and Open Access to Information on the Internet**
   Sponsor: Google Focus Program
   Investigator(s): Nick Feamster (PI), Wenke Lee, Mustaque Ahamad, Patrick Traynor, Henry Owen, Ellen Zegura, Zvi Galil

Amount: *$1,000,000 for 2 years*
Awarded: November 2011

9. **Dynamic-attribute-based Disclosure of Health Information in Emergency Care Scenarios**
Sponsor: Health Systems Institute (HSI) Seed Grant Program
Investigator(s): Doug Blough (PI), Mustaque Ahamad, Patrick Traynor and Jim Jose
Amount: *$50,000 over 1 year*
Awarded: August 2009

10. **Federal Cyber Service Scholarships at Georgia Tech**
Sponsor: NSF SFS Scholarships
Investigator(s): Seymour Goodman (PI), Patrick Traynor
Amount: *$1,250,682 over 5 years*
Awarded: June 2009

11. **Security for IMS-Enabled Converged Applications**
Sponsor: US Department of Defense
Investigator(s): Mustaque Ahamad (PI), Patrick Traynor (PI), Michael Hunter, Russ Clark
Amount: *$146,121 for 1 year*
Awarded: August 2008

2. Pending

Removed for external version.

I. Research Honors and Awards

1. Fellow, Center for Financial Inclusion at Accion, 2017.
2. Sloan Research Fellow, Alfred P. Sloan Foundation, 2014.
3. Best Paper, The ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec); Budapest, Hungary, 2013.
4. Best Student Paper, The Information Security Conference (ISC); Passau, Germany, 2012
5. Lockheed Inspirational Young Faculty Award, 2012
6. Best Demo, "Is Browsing the Internet on Your Mobile Phone Secure?" Chaitrali Amrutkar (Ph.D Advisee), CoC Research Day, 2011
7. Best Poster, "(sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers" Arunabh Verma, Henry Carter (MS, Ph.D Advisees), CoC Research Day, 2011
8. National Science Foundation CAREER Award, 2010
9. Pennsylvania State University Alumni Association Dissertation Award, 2007
10. Pennsylvania State University CSE Graduate Research Assistant Award, 2007
11. AT&T Wireless Fellowship, 2005

III. SERVICE

A. Professional Activities

A.1. Memberships and Activities in Professional Societies

1. Senior Member, Association for Computing Machinery (ACM)
2. Senior Member, Institute of Electrical and Electronics Engineers (IEEE)
3. Member, USENIX Advanced Computing Systems Association (USENIX)

A.2. Conference Committee Activities

1. Program co-Chair, *IEEE Symposium on Security and Privacy (OAKLAND)*: 2023, 2024
2. Program co-Chair, *USENIX Security Symposium (SECURITY)*: 2019
3. Program co-Chair, *Network and Distributed System Security Symposium (NDSS)*: 2017, 2018
4. Program Chair, *USENIX Workshop on Offensive Technologies (WOOT)*: 2016
5. Program Chair, *ACM Conference on Wireless Network Security (WiSec)*: 2014
6. Program Co-Chair, *Annual Computer Security Applications Conference (ACSAC)*: 2012, 2013
7. Program Chair, *USENIX Workshop on Hot Topics in Security (HotSec)*: 2012
8. Chair Invited Talks Committee, *USENIX Security Symposium (SECURITY)*: 2014
9. Workshops Chair, *IEEE Conference on Communications and Network Security (CNS)*: 2016
10. Program Committee, *USENIX Security Symposium (SECURITY)*: 2008, 2009, 2010, 2013, 2015-2018, 2020-2022
11. Program Committee, *IEEE Symposium on Security and Privacy (OAKLAND)*: 2009-2014, 2022.
12. Program Committee, *ACM Conference On Computer and Communications Security (CCS)*: 2009, 2013-2015, 2017
13. Program Committee, *Network and Distributed System Security Symposium (NDSS)*: 2010, 2013-2016, 2020-2021
14. Program Committee, *IEEE European Symposium on Security and Privacy (Euro S&P)*: 2016
15. Program Committee, *Annual Computer Security Applications Conference (ACSAC)*: 2008, 2009, 2010, 2011, 2015
16. Program Committee, *ACM Conference on Wireless Network Security (WiSec)*: 2009, 2010, 2013, 2015-2021
17. Program Committee, *International Conference on Financial Cryptography and Data Security (FC)*: 2010, 2013
18. Program Committee, *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*: 2016.
19. Program Committee, *ICST Conference on Security and Privacy in Communication Networks (SecureComm)*: 2009, 2010
20. Program Committee, *Privacy Enhancing Technologies Symposium (PETS)*: 2015, 2016

21. Program Committee, *International World Wide Web Conference (WWW)*: 2016
22. Program Committee, *USENIX Workshop on Hot Topics in Security (HotSec)*: 2011
23. Program Committee, *ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MOBIHELD)*: 2010
24. Program Committee, *International Workshop on Mobile Security (WMS)*: 2010
25. Program Committee, *European Symposium on Research in Computer Security (ESORICS)*: 2009, 2011
26. Program Committee, *IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS)*: 2009, 2010
27. Program Committee, *Information Security Conference (ISC)*: 2010
28. Program Committee, *IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT)*: 2009
29. Program Committee, *Computer Security Architecture Workshop (CSAW)*: 2008
30. Program Committee, *IWCMC Computer and Network Security Symposium*: 2009
31. Program Committee, *IARIA International Conference on Internet Monitoring and Protection (ICIMP)*: 2009
32. Program Committee, *IEEE Workshop on Network Security and Privacy (NSP)*: 2008
33. Program Committee, *IEEE International Workshop on Wireless and Sensor Networks Security (WSNS)*: 2008, 2009
34. Program Committee, *IEEE Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC)*: 2008
35. Program Committee, *ACM Conference on Computer and Communications Security, Industry and Government Track (CCS I&G)*: 2006, 2007
36. Program Committee, *Workshop on Secure Network Protocols (NPSec)*: 2006
37. Program Committee, *International Conference on Information Systems Security (ICISS)*: 2006, 2009, 2010
38. Program Committee, *IEEE LCN Workshop on Network Security (WNS)*: 2006, 2007, 2008

B. On-Campus Committees

B.1. University of Florida

1. Member, Computer and Information Science and Engineering Steering Committee, 2015-2017.
2. Member, Graduate Recruiting Committee, 2015-2017.
3. Chair, Computer and Information Science and Engineering Industrial Advisory Board, 2014-2015.

B.2. Georgia Tech

1. Member, Massive Open Online Master's (MOOMS) Investigation Committee, 2012-2013.
2. Chair, School of Computer Science Ph.D. Review Committee, 2012.
3. Member, School of Computer Science Ph.D Review Committee, 2011.
4. Faculty Advisor, Grey H@T - Georgia Tech Undergraduate Security Club, 2011-2014.
5. Member, School of Computer Science Ph.D. Review Committee, 2011.
6. Member, School Advisory Committee, School of Computer Science, 2011-2013.
7. Member, School of Computer Science Chair Recruiting Committee, 2011.
8. Member, School of Computer Science Faculty Recruiting Committee, 2010, 2011.
9. Chair, College of Computing Ph.D. Welcome Weekend Committee, 2009, 2010, 2011 (co-chair).
10. Member, College of Computing Ph.D. Recruiting Committee, 2009.
11. Member, Georgia Tech Computer and Network Usage Security Policy (CNUSP) Evaluation Group, 2009.

C. Special Assignments

None.

D. Ph.D. Examining Committees

Ph.D. Examining Committees

1. Bradley Reaves, Department of Computer and Information Science and Engineering, University of Florida, Summer 2017.
   Advisor: Professor Patrick Traynor.
2. Adam Bates, Department of Computer and Information Science and Engineering, University of Florida, Spring 2016.
   Advisor: Professor Kevin Butler.
3. Benjamin Mood, Department of Computer and Information Science and Engineering, University of Florida, Spring 2016.
   Advisor: Professor Kevin Butler.
4. Henry Carter, College of Computing, Georgia Tech, Fall 2015.
   Advisor: Professor Patrick Traynor.
5. David Dewey, College of Computing, Georgia Tech, Fall 2015.
   Advisor: Professor Patrick Traynor.
6. Lateef Yusuf, College of Computing, Georgia Tech, Spring 2014.
   Advisor: Professor Umakishore Ramachandran.
7. Chaitrali Amrutkar, College of Computing, Georgia Tech, Summer 2013.
   Advisor: Professor Patrick Traynor.
8. Long Lu, College of Computing, Georgia Tech, Summer 2013.
   Advisor: Professor Wenke Lee.

9. Manos Antonakakis, College of Computing, Georgia Tech, Summer 2012.
   Advisor: Professor Wenke Lee.

10. Junjie Zhang, College of Computing, Georgia Tech, Summer 2012.
    Advisor: Professor Wenke Lee.

11. Italo Dacosta, College of Computing, Georgia Tech, Summer 2012.
    Advisor: Professor Patrick Traynor.

12. Virendra Kumar, College of Computing, Georgia Tech, Summer 2012.
    Advisor: Professor Alexandra Boldyreva.

13. Anirudh Ramachandran, College of Computing, Georgia Tech, Summer 2011.
    Advisor: Professor Nick Feamster.

14. Vijay Balasubramaniyan, College of Computing, Georgia Tech, Summer 2011.
    Advisor: Professor Mustaque Ahamad.

15. Kapil Singh, College of Computing, Georgia Tech, Summer 2011.
    Advisor: Professor Wenke Lee.

16. Abhinav Srivastava, College of Computing, Georgia Tech, Summer 2011.
    Advisor: Professor Jon Giffin.

17. Adam O'Neill, College of Computing, Georgia Tech, Summer 2010.
    Advisor: Professor Alexandra Boldyreva.

18. David Cash, College of Computing, Georgia Tech, Fall 2009.
    Advisor: Professor Alexandra Boldyreva.

External Member of Ph.D. Research Committee

None.

External Member of Ph.D. Examining Committee

1. Shannon Eggers, Department of Materials Sciences and Engineering - Nuclear Engineering Program, University of Florida, Fall 2016.
   Advisor: Professor Kelly Jordan.

2. Ed Carlisle, Department of Electrical and Computer Engineering, University of Florida, Summer 2016.
   Advisor: Professor Alan George.

3. Claudio Marforio, Department of Computer Science, Swiss Federal Institute of Technology Zurich (ETH Zurich), Fall 2015.
   Advisor: Professor Srdjan Capkun.

4. Nils Ole Tippenhauer, Department of Computer Science, Swiss Federal Institute of Technology Zurich (ETH Zurich), Spring 2012.
   Advisor: Professor Srdjan Capkun.

5. Bongkyoung Kwon, School of Electrical and Computer Engineering, Georgia Tech, Summer 2009.
   Advisor: Professor John Copeland.

Ph.D. Thesis Proposal Committees

1. Bradley Reaves, Department of Computer and Information Science and Engineering, Spring 2016.
   Advisor: Professor Patrick Traynor.
2. Maliheh Shirvanian, University of Alabama, Birmingham, Spring 2016.
   Advisor: Professor Nitesh Saxena.
3. Benjamin Mood, Department of Computer and Information Science and Engineering, Fall 2015.
   Advisor: Professor Kevin Butler.
4. Adam Bates, Department of Computer and Information Science and Engineering, Fall 2015.
   Advisor: Professor Kevin Butler.
5. Lateef Yusuf, College of Computing, Georgia Tech, Spring 2013.
   Advisor: Professor Umakishore Ramachandran.
6. Long Lu, College of Computing, Georgia Tech, Spring 2013.
   Advisor: Professor Wenke Lee.
7. Chaitrali Amrutkar, College of Computing, Georgia Tech, Fall 2012.
   Advisor: Professor Patrick Traynor.
8. Junjie Zhang, College of Computing, Georgia Tech, Fall 2011.
   Advisor: Professor Wenke Lee.
9. Italo Dacosta, College of Computing, Georgia Tech, Fall 2011.
   Advisor: Professor Patrick Traynor.
10. Manos Antonakakis, College of Computing, Georgia Tech, Fall 2011.
    Advisor: Professor Wenke Lee.
11. Abhinav Srivastava, College of Computing, Georgia Tech, Spring 2011.
    Advisor: Professor Jon Giffin.
12. Vijay Balasubramaniyan, College of Computing, Georgia Tech, Fall 2010.
    Advisor: Professor Mustaque Ahamad.
13. Kapil Singh, College of Computing, Georgia Tech, Fall 2010.
    Advisor: Professor Wenke Lee.
14. Anirudh Ramachandran, College of Computing, Georgia Tech, Fall 2010.
    Advisor: Professor Nick Feamster.
15. Adam O'Neill, College of Computing, Georgia Tech, Spring 2010.
    Advisor: Professor Alexandra Boldyreva.
16. David Cash, College of Computing, Georgia Tech, Spring 2009.
    Advisor: Professor Alexandra Boldyreva.

Ph.D. Qualifying Exam Committees—Georgia Tech

1. Byoungyoung Lee, College of Computing, Georgia Tech, Spring 2013.
   Advisor: Professor Wenke Lee.
2. Yizheng Chen, College of Computing, Georgia Tech, Spring 2013.
   Advisor: Professor Wenke Lee.

3. Xinyu Xing, College of Computing, Georgia Tech, Spring 2013.
   Advisor: Professor Wenke Lee.
4. Brad Reaves, College of Computing, Georgia Tech, Spring 2013.
   Advisor: Professor Patrick Traynor.
5. Chaz Lever, College of Computing, Georgia Tech, Spring 2013.
   Advisor: Professor Patrick Traynor.
6. Terry Nelms, College of Computing, Georgia Tech, Spring 2012.
   Advisor: Professors Mustaque Ahamad and Roberto Perdesci.
7. Saurabh Chakradeo, College of Computing, Georgia Tech, Spring 2012.
   Advisor: Professor Patrick Traynor.
8. Henry Carter, College of Computing, Georgia Tech, Spring 2012.
   Advisor: Professor Patrick Traynor.
9. David Dewey, College of Computing, Georgia Tech, Spring 2012.
   Advisor: Professor Jon Giffin.
10. Chaitrali Amrutkar, College of Computing, Georgia Tech, Fall 2011.
    Advisor: Professor Patrick Traynor.
11. Yacin Nadji, College of Computing, Georgia Tech, Fall 2011.
    Advisor: Professor Wenke Lee.
12. Yogesh Mundada, College of Computing, Georgia Tech, Fall 2011.
    Advisor: Professor Nick Feamster.
13. Hyojoon Kim, College of Computing, Georgia Tech, Fall 2011.
    Advisor: Professor Nick Feamster.
14. Ikpeme Erete, College of Computing, Georgia Tech, Spring 2011.
    Advisor: Professor Alex Orso.
15. Chaitrali Amrutkar, College of Computing, Georgia Tech, Spring 2011.
    Advisor: Professor Patrick Traynor.
16. Brendan Dolan-Gavitt, College of Computing, Georgia Tech, Spring 2011.
    Advisor: Professor Wenke Lee and Professor Jon Giffin.
17. Sam Burnett, College of Computing, Georgia Tech, Fall 2010.
    Advisor: Professor Nick Feamster.
18. Cong Shi, College of Computing, Georgia Tech, Fall 2010.
    Advisor: Professor Mostafa Ammar and Professor Ellen Zegura.
19. Partha Kanuparthy, College of Computing, Georgia Tech, Fall 2010.
    Advisor: Professor Constantine Dorvolis.
20. Long Lu, College of Computing, Georgia Tech, Spring 2010.
    Advisor: Professor Wenke Lee.
21. Virendra Kumar, College of Computing, Georgia Tech, Spring 2009.
    Advisor: Professor Alexandra Boldyreva.
22. Frank Park, College of Computing, Georgia Tech, Spring 2009.
    Advisor: Professor Patrick Traynor.

23. Italo Dacosta, College of Computing, Georgia Tech, Fall 2008.
Advisor: Professor Mustaque Ahamad and Professor Patrick Traynor.

24. Adam O'Neill, College of Computing, Georgia Tech, Fall 2008.
Advisor: Professor Alexandra Boldyreva.

E. External Member of M.S. Examining Committee

M.S. Thesis Defense Committees None.

F. Consulting and Advisory Appointments

1. Skim Reaper, *Co-Founder and CEO*, 2019-Present.
2. CryptoDrop Anti-Ransomware, *Co-Founder and CEO*, 2017-2018.
3. Pindrop Security, *Research Fellow and Co-Founder*, Spring 2012 - Spring 2014.
4. United States Army (via US Falcon), *Information Assurance Officer Training Program*, Spring 2010.
5. 3G Americas, *Characterizing the Limitations of Third-Party EAS over Cellular Text Messaging Systems*, Fall 2008.

G. Civic Activities

None.

IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION

A. Honors and Awards

1. Fellow, Kavli Foundation, 2017.
2. Fellow, Center for Financial Inclusion at Accion, 2016.
3. Sloan Research Fellow, Alfred P. Sloan Foundation, 2014.

B. Invited Conference Session Chairmanships

1. Session Chair, *Work-in-Progress* at the *USENIX Security Symposium (SECURITY)*, 2016.
2. Session Chair, *Mobile Security* at the *USENIX Security Symposium (SECURITY)*, 2013.
3. Poster Chair, *USENIX Security Symposium (SECURITY)*, 2010, 2011.
4. Session Chair, *Privacy and Anonymity* at the *USENIX Workshop on Hot Topics in Security (HotSec)*, 2011.
5. Session Chair, *Security of Authentication and Protection Mechanisms* at the *IEEE Symposium on Security & Privacy (OAKLAND)*, 2011.
6. Session Chair, *Information Abuse* at the *IEEE Symposium on Security & Privacy (OAKLAND)*, 2010.
7. Session Chair, *RFID Security* at the *ACM Conference on Computer and Communications Security (CCS)*, 2009.
8. Session Chair, *Browser Security Session* at the *USENIX Security Symposium (SECURITY)*, 2009.
9. Session Chair, *Information Security Session* at the *IEEE Symposium on Security and Privacy (OAKLAND)*, 2009.
10. Session Chair, *Work-in-Progress* at the *IEEE Symposium on Security and Privacy (OAKLAND)*, 2009.
11. Session Chair, *Work/Opinions-in-Progress* at the *ISOC Network and Distributed Systems Security (NDSS) Symposium*, 2009.
12. Session Chair, *Privacy Session* at the *USENIX Security Symposium (SECURITY)*, 2008.

C. Professional Registration

None.

D. Patents

1. Patrick G. Traynor, Christian Peeters, Bradley G. Reaves, Hadi Abdullah, Kevin Butler, Jasmine Bowers, Walter N. Scaife, "Detecting SS7 Redirection Attacks With Audio-Based Distance Bounding", United State Patent # 11,265,717, Filed March 2019, Issued March 2022.
2. Patrick G. Traynor, Logan E. Blue, Luis Vargas, "Method and Apparatus for Differentiating Between Human and Electronic Speaker for Voice Interface Security", United State Patent # 11,176,960, Filed June 2019, Issued November 2021.
3. Patrick G. Traynor, Bradley G. Reaves, Logan E. Blue Practical End-to-End Cryptographic Authentication for Telephony Over Voice Channels, United State Patent # 11,329,831, Filed November 2018, Issued May 2022.

4. Walter Nolen Scaife, Patrick G. Traynor and Christian Peeters, "Payment Card Overlay Skimmer Detection", United States Patent # 10,496,914, Filed October 2017, Issued December 2019. (See also # 10,936,928)
5. Patrick G. Traynor, David P. Arnold, Walter Nolen Scaife, Christian Peeters, and Camilo Valez Cuervo, "Detecting counterfeit magnetic stripe cards using encoding jitter", United States Patent # 10,803,261, Filed May 2017, Issued October 2020.
6. Patrick G. Traynor, Bradley Reaves, Logan Blue, Luis Vargas, Hadi Abdullah, and Thomas Shrimpton, "Identity and content authentication for phone calls", United States Patent # 10,764,043, Filed Apr 2017, Issued September 2020.
7. Walter Nolen Scaife, Henry Carter, Patrick G. Traynor and Kevin R. B. Butler. "Malware Detection Through User Data Transformation Monitoring", United States Patent # 10,685,114. Filed September 2015, Issued June 2020.
8. Vijay A. Balasubramaniyan, Mustaque Ahamad, Patrick G. Traynor. "Using Single-Ended Audio Features to Automatically Determine Voice Call Provenance", United States Patent, #9,037,113 June 2010, Issued May 2015. (See also #9,516,497 and #10,523,809)
9. Patrick G. Traynor, Byungsuk Kim and Farooq Anjum. "Secure Localization for 802.11 Networks with Fine Granularity", United States Patent, #8,107,400, Filed July 2008, Issued January 2012.

E. Editorial and Reviewer Work for Technical Journals and Publishers

Associate Editor:

- *ACM Transactions on Information and System Security (TISSEC)* 2015-present

Guest Editor:

Journals

- *IEEE Security and Privacy Magazine (S&P)* 2013

Reviewer for:

Journals

- *ACM Transactions on Information and System Security (TISSEC)* 2008, 2009, 2010, 2011, 2012, 2013
- *IEEE Transactions on Dependable and Secure Computing (TDSC)* 2012, 2013
- *IEEE Security and Privacy Magazine (S&P)* 2010, 2011
- *Communications of the ACM (CACM)* 2010
- *Journal of Anesthesia & Analgesia* 2009
- *IEEE Transactions on Mobile Computing (TMC)* 2008, 2010, 2011, 2012, 2013
- *IEEE Transactions on Internet Technology (TOIT)* 2009, 2010
- *ACM Mobile Computing and Communications Review (MC2R)* 2008
- *IEEE/ACM Transactions on Networking (TON)* 2007, 2008
- *Journal of Pervasive and Mobile Computing (PMC)* 2009, 2010

- *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 2005, 2009, 2010
- *IEEE Transactions on Computers (TOC)* 2010
- *Journal of Security and Communication Networks (SCN)* 2008
- *IEEE Communications Letters (CL)* 2007, 2009
- *IEEE Transactions on Wireless Communications (TWC)* 2007
- *Pervasive and Mobile Computing (PMC)* 2007
- *IEEE Transactions on Software Engineering (TSE)* 2007, 2008
- *Journal of Wireless Networks (WiNet)* 2006, 2007, 2008, 2009
- *Journal of Wireless Communications and Mobile Computing* 2006
- *ACM Computing Surveys (ACMCS)* 2006
- *Information Processing Letters (IPL)* 2006
- *IEEE Transactions on Very Large Scale Integration Systems (TVLSIS)* 2006

Conferences and Workshops

- *ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2011
- *ACM Conference on Computer and Communications Security (CCS)*, 2008, 2011
- *IEEE Symposium on Security and Privacy (OAKLAND)* 2007, 2008
- *Computer Security Foundations (CSF)*, 2011
- *IFIP Conference on Data and Applications Security (DBSec)* 2008
- *Financial Cryptography (FC)* 2007, 2008
- *International Conference on VLSI Design (VLSI)* 2007
- *Annual Computer Security Applications Conference (ACSAC)* 2005, 2006, 2007
- *USENIX Workshop on Hot Topics in Security (HotSec)* 2007
- *International Conference on Information Systems Security (ICISS)* 2007
- *IEEE International Conference on Computer Engineering & Systems (ICCES)* 2007
- *International Workshop on Security (IWSec)* 2006, 2007
- *USENIX Security Symposium (SECURITY)* 2006, 2007
- *IEEE Sarnoff Symposium (SARNOFF)* 2007
- *International Conference on New Technologies, Mobility and Security (NTMS)* 2007
- *IEEE Infocom (INFOCOM)* 2007
- *Network and Distributed System Security Symposium (NDSS)* 2007
- *International Workshop on Storage Security and Survivability (IWSSS)* 2006
- *ACM Conference on Computer and Communications Security (CCS)* 2006

- *IEEE GLOBECOM (GLOBECOM)* 2006
- *International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS)* 2006
- *IFIP Conference on Data and Applications Security (DBSec)* 2006
- *Emerging Trends in Information and Communications Security (ETRICS)* 2006
- *International Conference on Applied Cryptography and Network Security (ACNS)* 2006
- *ACM Symposium on Access Control Models and Technology (SACMAT)* 2006
- *IEEE Conference on Communication Systems Software & Middleware (COMSWARE)* 2006
- *International Conference on Cryptology in India (IndoCrypt)* 2005
- *IEEE Symposium on New Frontiers in Dynamic Spectrum Access (DySPAN)* 2005
- *European Symposium on Research in Computer Security (ESORICS)* 2005

F. Expert Witness Services

1. *Natalie Delgado, et al. v. Meta Platforms Inc., Case No.: 23-cv-04181 (N.D. Cal.):* Expert witness for the Defense (via Gibson, Dunn & Crutcher, LLP). *October 2025 - Present*.
2. *Google LLC v Headwater Research LLC:* Expert witness for the Plaintiff for Inter Partes Review (via Wolf, Greenfield & Sacks, P.C.). *August 2025 - Present*.
3. *Amazon v Headwater Research LLC:* Expert witness for the Plaintiff for Inter Partes Review (via Perkins Coie, LLP). *August 2025 - Present*.
4. *HBCU Messaging US LP v. Apple, Inc. et al., Civil No. 1:24-cv-1199,:* Expert witness for the Defense (via Fish & Richardson, LLP). *June 2025 - Present*.
5. *PACid v. Citibank, N.A., 1:24-cv-00272-DAE (W.D. Tex.):* Expert witness for the Defense (via Troutman Pepper Locke LLP). *April 2025 - November 2025*.
6. *Facebook Inc. Derivative Litigation., Consolidated C.A. No. 2018-0307-JTL (Del. Ch.):* Expert witness for the Defense (via Wachtell, Lipton, Rosen & Katz, LLP) *January 2025 - July 2025*.
7. *Averon US, Inc. vs AT&T Inc., Case No. 1:22-cv-01341-TMH:* Expert witness for the Defense (via O'Melveny & Myers, LLP). *November 2024 - September 2025*.
8. *RightQuestion, LLC v. AT&T Inc. et al.,., Case No. 2-24-cv-00094 -JRG:* Expert witness for the Defense (via Duane Morris, LLP). *September 2024 - Present*.
9. *ByteDance Ltd. vs CellSpin Soft, Inc.:* Expert witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *February 2024 - October 2024.*
10. *Bank of America, N.A., vs PACid:* Expert witness for the Plaintiff for Inter Partes Review (via McNish PLLC). *December 2023 - March 2024*.
11. *Microsoft vs Proxense, LLC:* Expert witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *November 2023 - July 2025*.
12. *Samsung vs Headwater Research LLC:* Expert witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *August 2023 - October 2025*.

13. *Advanced Coding Technologies LLC v. ByteDance PTE. Ltd., and TikTok PTE. Ltd.:* Expert witness for the Defense for Non-Infringement (via Fish & Richardson, LLP). *July 2023 - September 2023*.

14. *Epic Games, Inc. & Anor v Google LLC & Ors - Federal Court of Australia Proceeding NSD 190 of 2021:* Expert witness for the Defendant (via Corrs Chambers Westgarth). *January 2023 - June 2024*.

15. *Rubin vs KAHOOT! ASA and KAHOOT EDU:* Expert witness for the Defendant for Inter Partes Review (via Vasquez Benisek & Lindgren, LLP). *December 2022 - June 2024*.

16. *Telefonaktiebolaget LM Ericsson vs Apple, Inc:* Expert witness for the Defendant, Non-Infringement and Invalidity (via WilmerHale LLP). *February 2022 - December 2022*.

17. *Wepay Global Payments, LLC v. Bank of America N. A.,:* Expert Witness for the Defendant (via WilmerHale LLP) *September 2022 - November 2022*.

18. *Apple vs. R.N Nehushtan Trust Ltd.:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *August 2022 - June 2023*.

19. *Apple/Microsoft vs. Zipit Wireless:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *May 2021 - November 2022*.

20. *Blackberry Inc v MobileIron, Inc:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *January 2021 - March 2021*.

21. *Apple Inc v Seven Networks, LLC:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *August 2019 - November 2020*.

22. *mSIGNIA, Inc. v. InAuth, Inc.:* Expert Witness for the Defendant for Inter Partes Review, Non-Infringement and Invalidity, Trade Secrets (via Quinn Emanuel Urquhart and Sullivan, LLP). *October 2017 - December 2018*.

23. *Huawei v. T-Mobile:* Expert Witness for the Defendant for Non-Infringement (via WilmerHale LLP, Alston & Bird LLP) *June 2016 - December 2017*.

24. *Telefonaktiebolaget LM Ericsson v Apple:* Expert Witness for the Defendant for Non-Infringement, Invalidity (via WilmerHale LLP). *June 2015 - December 2015*.

25. *Mayfonk v Nike:* Expert Witness for the Plaintiff for Infringement/Trade Secrets (via Paul Hastings). *June 2015 - November 2015*.

26. *Maxim Integrated Products v Bank of the West:* Expert Witness for the Defendant for Non-Infringement (via Paul Hastings LLP). *January 2014 - August 2014*.

27. *Maxim Integrated Products v Comerica Inc, et al:* Expert Witness for the Defendant for Non-Infringement (via McKenna, Long & Aldridge LLP). *June 2014 - August 2014*.

28. *William Grecia v. Apple Inc. et al:* Expert Consultant for the Defendant for Invalidity (via Kirkland & Ellis LLP). *July 2014 - August 2014*.

29. *Intertrust Technologies Corp. v. Apple Inc.:* Expert Consultant for Defendant for Invalidity and Non-Infringement (via Kirkland & Ellis LLP). *October 2013 - February 2014*.

30. *Maxim Integrated Products v KeyCorp Bank:* Expert Witness for the Defendant for Non-Infringement (via Calfee, Halter & Griswold LLP) *April 2013 - June 2013*.

31. *Intellectual Ventures LLC vs. Check Point; et al.:* Expert Consultant for the Plantiff for Infringement (via Susman Godfrey LLP), *October 2012 - February 2015*.

V. OTHER CONTRIBUTIONS

A. Seminar Presentations (Invited Papers and Talks at Meetings and Symposia)

1. Keynote: Well, It Worked on My Computer: Reproducibility in Computer Security Research. Learning from Authoritative Security Experiment Results (LASER) Workshop, December 2024. École Polytechnique Fédérale de Lausanne (EPFL, Switzerland).

2. Social Engineering and Two-Factor Authentication. Cyber Security Training Eastern Indonesia, August 2024. Financial Innovation Lab (EIFIL) (Denpasar, Bali, Indonesia).

3. DFS Security and Mobile Money Analysis. Cyber Security Training Eastern Indonesia, August 2024. Financial Innovation Lab (EIFIL) (Denpasar, Bali, Indonesia).

4. Keynote: Well, It Worked on My Computer: Reproducibility in Computer Security Research. EPFL Summer Research Institute (SURI), July 2024. École Polytechnique Fédérale de Lausanne (EPFL, Switzerland).

5. Humans vs The Computer Interfaces: The Challenge of Separating Deepfakes/Bots from People. North Central Florida Institute of Internal Auditors (IIA), May 2024.

6. Humans vs The Computer Interfaces: The Challenge of Separating Deepfakes/Bots from People. UF Quest 2: Siri is my Superpower: Communicating with AI, March 2024. University of Florida.

7. Humans vs The Computer Interfaces: The Challenge of Separating Deepfakes/Bots from People. Federal Information Integrity Research and Development (FIIRD) Interworking Group (IWG), March 2024. via NITRD, OSTP.

8. AI driven voice cloning scams. Discussion at the White House with Anne Neuberger (Deputy National Security Advisor for Cyber and Emerging Technologies), Jessica Rosenworcel (Chair of the Federal Communications Commission) and Lina Khan (Chair of the Federal Trade Commission), January 2024. Lead Academic facilitator.

9. Keynote: Well, It Worked on My Computer: Reproducibility, Tech Transfer, and Computer Security Research. National Science Foundation Secure and Trustworthy Cyberspace (SaTC) Vision 2.0 Workshop, March 2023. University of Texas at Dallas.

10. Humans vs The Computer Interfaces: Separating Deepfakes/Bots from People Using Psychoacoustics. UCLA Electrical and Computer Engineering Distinguished Seminar, February 2023. University of California, Los Angeles.

11. Keynote: Exploiting the Gaps Between Human and Machine Understanding of Audio: Frameworks, Attacks, and Defenses. ISCA Symposium on Security and Privacy in Speech Communication (SPSC), November 2021. Virtual.

12. The State of Voice Cloning Technology. Federal Trade Commission (FTC) Workshop on Voice Cloning Technologies, January 2020. Washington, DC.

13. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. North Carolina State University Department of Computer Science Colloquium, January 2020. Raleigh, NC.

14. Moving from research to practice: How to maximize the impact of SaTC projects. National Science Foundation Secure and Trustworthy Cyberspace (SaTC) PI Meeting, October 2019. Alexandria, VA.

15. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. Purdue University Computer Science Excellence Lecture Series, October 2019. West Lafayette, IN.

16. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. Bank of America - Colloquium Series, March 2019. Charlotte, NC.

17. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. CISPA – Helmholtz Center for Information Security, Saarland University, February 2019. Saarbrücken, Germany.

18. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. University of Maryland - Distinguished Colloquium, February 2019. College Park, MD.

19. Responsible Finance for the Digital Client. Foromic Conference, October 2018. Barranquilla, Colombia.

20. Panel: Authentication Challenges for New Interfaces, Devices, and Wireless Networks. ACM Conference on Security and Privacy in Wireless and Mobile Networks, June 2018. Stockholm, Sweeden.

21. Sonar: Detecting SS7 Redirection Attacks Via Call Audio-Based Distance Bounding. CyberSecurity@KAIST Workshop - KAIST, June 2018. Daejon, South Korea.

22. Why Caller-ID Spoofing Is So Easy (and Why End-To-End Solutions Are the Way Forward). IEEE Workshop on Technology and Consumer Protection (ConPro'18), May 2018. San Francisco, CA.

23. Panel: The Future of Cybersecurity. SEC Academic Conference - Auburn University, May 2018. Auburn, AL.

24. Sound Principles: Verifying Voice Commands in an IoT World. IoT Security Workshop - Aalto University, September 2017. Helsinki, Finland.

25. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Eurecom Institute, September 2017. Sophia Antipolis, France.

26. Panel: Infrastructure Stability. ITU-T Focus Group Digital Financial Services, December 2016. Geneva, Switzerland.

27. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. ETH Zurich, December 2016. Zurich, Switzerland.

28. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. University of Richmond, October 2016. Richmond, Virginia.

29. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Indiana University, September 2016. Bloomington, Indiana.

30. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Aalto University Computer Science Department Forum, August 2016. Helsinki, Finland.

31. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. KAIST Information Security Seminar - Korean Advanced Institute of Science and Technology, June 2016. Daejeon, South Korea.

32. Updated Mobile Money Vulnerability Report. International Telecommunications Union Digital Financial Services Working Group Workshop, May 2016. Washington, DC.

33. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. UF Eye Opener Discovery Breakfast - University of Florida, May 2016. Gainesville, FL.

34. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Illinois Science of Security (SoS) Lablet Speaker Series - University of Illinois, Urbana-Champaign, April 2016. Urbana-Champaign, Illinois.

35. New Trends in Cybersecurity:Vulnerabilities in Branchless Banking Systems. United States Departments of State and Justice - Cybersecurity and Cybercrime Workshop for Lusophone Africa, September 2015. Maputo, Mozambique.

36. New Trends in Cybersecurity:Vulnerabilities in Branchless Banking Systems. United States Departments of State and Justice - ECCAS Cybersecurity and Cybercrime Workshop, August 2015. Kinshasa, Democratic Republic of Congo.

37. Chasing Telephony Security: Where the Wild Things... Are? University of Florida - Department Colloquium, January 2014. Gainesville, FL.

38. Chasing Telephony Security: Where the Wild Things... Are? Verizon Wireless RNC/Data Center, October 2013. Alpharetta, GA.

39. Chasing Telephony Security: Where the Wild Things... Are? University of Waterloo - CrySP Speaker Series on Privacy, October 2013. Waterloo, ON, Canada.

40. Analyzing Malicious Traffic in Cellular Networks. GSM Association's (GSMA) Mobile Malware Community Workshop, July 2013. Mountain View, CA.

41. Threats to Mobile Devices. US Federal Trade Commission (FTC) Public Forum - Invited Speaker, June 2013. Washington, D.C.

42. Chasing Telephony Security: Where the Wild Things... Are? University of Wisconsin - Madison, Security Seminar, March 2013. Madison, WI.

43. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. Invited Talk: Centre for Secure Information Technologies (CSIT) Queen's University, March 2013. Belfast, Northern Ireland.

44. Chasing Telephony Security: Where the Wild Things... Are? Stanford Security Seminar, March 2013. Stanford, CA.

45. Chasing Telephony Security: Where the Wild Things... Are? University of California, Berkeley, Security Group, March 2013. Berkeley, CA.

46. Chasing Telephony Security: Where the Wild Things... Are? Carnegie Mellon University CyLab Seminar, February 2013. Pittsburgh, PA.

47. Chasing Telephony Security: Where the Wild Things... Are? University of Oregon Department of Computer Science Colloquium, November 2012. Eugene, OR.

48. Chasing Telephony Security: Where the Wild Things... Are? University of Washington Department of Electrical Engineering, Network Security Lab (NSL): Invited Talk, November 2012. Seattle, WA.

49. Needles and Haystacks: Digging for Ground Truth on Mobile Malware. ZISC Workshop on Secure Mobile and Cloud Computing, ETH Zurich, June 2012. Zurich, Switzerland.

50. Panel: Advice for Early Career Faculty. CRA Career Mentoring Workshop, February 2012. Washington, D.C.

51. Research Challenges in Cellular and Mobile Network Security. US-China Software Workshop (Co-Sponsored by NSF and NSFC), September 2011. Beijing, China.

52. Mobile Security: Understanding Risks to Critical Infrastructure. Invited Talk: US Department of State East African Workshop on Cyberspace Security, July 2011. Nairobi, Kenya.

53. Tomorrow's Issues: Solving the Mobile Security Threat. Invited Talk: Centre for Secure Information Technologies (CSIT) Queen's University, March 2011. Belfast, Northern Ireland.

54. PinDr0p: Using Single-Ended Audio Features to Determine Call Provenance. Invited Talk: MITRE Corporation, March 2011. Burlington, MA.

55. Defeating Session Hijacking Attacks with Disposable Web Credentials. Invited Talk: Facebook, February 2011. Palo Alto, CA.

56. Understanding the Disruptive Potential of Malware in Cellular Networks. Invited Talk: RSA Conference, February 2011. San Francisco, CA.

57. Panel: Voice Security – Now Just a False Sense of Security and Privacy. Invited Panelist: Mobile Security Symposium, February 2011. San Francisco, CA.

58. Understanding the Disruptive Potential of Malware in Cellular Networks. Invited Talk: Concordia University, May 2010. Montreal, QC, Canada.

59. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Qualcomm Research, March 2010. San Diego, CA.

60. Privacy and Security Concerns for Personal and Mobile Health Devices. Invited Talk: Workshop to Set A Research Agenda for Privacy and Security of Healthcare Technologies, October 2009. Indianapolis, IN.

61. Considerations for EAS Over Cellular Text Messaging Services. 3G Americas Webinar, July 2009.

62. University Telephony Research Panel. Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM), July 2009.

63. The Evolving Mobile Landscape: Emerging Security Threats. Mobile Security eConference, SC Magazine, June 2008.

64. Characterizing the Impact of Rigidity on the Security of Cellular Networks. University of Washington, February 2009. Seattle, WA.

65. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Microsoft Research, February 2009. Redmond, WA.

66. Next Year's Problems. Secure Computing (SC) Magazine Webinar, November 2008.

67. Panel: Embedded Systems and their Increasing Impact on Infrastructure Security. Workshop on Embedded Systems Security (WESS), October 2008.

68. Can you DoS me now? Security Issues in Cellular Networks. Georgia Institute of Technology, September 2008. Atlanta, GA.

69. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Georgia Institute of Technology, April 2008. Atlanta, GA.

70. Characterizing the Impact of Rigidity on the Security of Cellular Networks. AT&T Research Labs, April 2008. Florham Park, NJ.

71. Characterizing the Impact of Rigidity on the Security of Cellular Networks. University of Arizona, March 2008. Tucson, AZ.

72. Cellular Networks Security Panel. USENIX Security Symposium, August 2007. Boston, MA.

73. malnets:Large-Scale Malicious Networks via Compromised Access Points. The Pennsylvania State University - ACM Club Invited Speaker, October 2006. State College, PA.

74. malnets:Large-Scale Malicious Networks via Compromised Access Points. The University of Michigan, October 2006. Ann Arbor, MI.

75. Exploiting Open Functionality in SMS-Capable Cellular Networks. The University of Michigan, October 2006. Ann Arbor, MI.

76. Exploiting Open Functionality in SMS-Capable Cellular Networks. High Technology Crime Investigation Association (HTCIA), September 2006. Pittsburgh, PA.

77. Trends in Security: Critical Engineering in the Large. Schlumberger Innovate IT! Workshop, May 2006. Cambridge, MA.

78. Exploiting Open Functionality in SMS-Capable Cellular Networks. InfraGard Pittsburgh Chapter General Meeting, September 2006. Pittsburgh, PA.

79. Exploiting Open Functionality in SMS-Capable Cellular Networks. InfraGard Pittsburgh Chapter General Meeting, September 2006. Pittsburgh, PA.

B. Special Activities

Presentations to Lay Media

1. How technology can fight digital fakery. The Babbage Podcast/The Economist https://shows.acast.com/theeconomistbabbage/episodes/babbage-how-to-detect-a-deepfake, January 2023.

2. Deepfake audio has a tell and researchers can spot it. Ars Technica https://arstechnica.com/information-technology/2022/09/researchers-use-fluid-dynamics-to-spot-deepfake-voices/, September 2022.

3. This security tool could help stop the problem of ransomware in its tracks. TheJournal.ie https://www.thejournal.ie/ransomware-researchers-stop-2875032-Jul2016/, July 2016.

4. Researchers Unleash Ransomware Annihilation. BankInfoSecurity - http://www.bankinfosecurity.com/researchers-unleash-ransomware-annihilation-a-9255, July 2016.

5. CryptoDrop Stops Ransomware by Stopping its Encryption. Security Intelligence - https://securityintelligence.com/news/cryptodrop-stops-ransomware-by-stopping-its-encryption/?utm_source=tfeed&utm_medium=twitter, July 2016.

6. Ransomware 'stopped' by new software. BBC - http://www.bbc.com/news/technology-36772461, July 2016.

7. Researchers create effective anti-ransomware solution. Help Net Security - https://www.helpnetsecurity.com/2016/07/12/anti-ransomware-solution/, July 2016.

8. Florida U boffins think they've defeated all ransomware. http://www.theregister.co.uk/2016/07/12/ransomware_defeated/, July 2016.

9. This Anti-Ransomware Tool Could Save You Hundreds of Pounds. Huffington Post - http://www.huffingtonpost.co.uk/entry/anti-ransomware-tool-save-hundreds-pounds_uk_57838beee4b0935d4b4b30ba, July 2016.

10. Researchers develop method to stop 100% of ransomware before it encrypts all files. Myce - http://www.myce.com/news/researchers-develop-method-stop-100-ransomware-encrypts-files-79873/, July 2016.

11. Desarrollan una solución para detener el ransomware. ComputerHoy - http://computerhoy.com/noticias/software/desarrollan-solucion-detener-ransomware-47972, July 2016.

12. Why your antivirus software can't stop ransomware. Futurity - http://www.futurity.org/ransomware-computer-files-1198242-2/, July 2016.

13. CryptoDrop Gives Users Hope to Prevent Ransomware Infections in the Future. Softpedia - http://news.softpedia.com/news/cryptodrop-gives-users-hope-to-prevent-ransomware-infections-in-the-future-506187.shtml, July 2016.

14. Could this be the answer to the ransomware threat?, Consumer Affairs. Consumer Affairs - https://www.consumeraffairs.com/news/could-this-be-the-answer-to-the-ransomeware-threat-071116.html, July 2016.

15. Extortion extinction: Researchers develop a way to stop ransomware. Phys.org - http://phys.org/news/2016-07-extortion-extinction-ransomware.html, July 2016.

16. Researchers Develop A Way To Stop Ransomware By Watching The Filesystem. Slashdot - https://yro.slashdot.org/story/16/07/08/2242244/researchers-develop-a-way-to-stop-ransomware-by-watching-the-filesystem, July 2016.

17. Mohul Ghosh. Trak.in - Digital Money Apps In India Are Unsafe and Unsecured – Researchers. http://trak.in/tags/business/2015/08/17/digital-money-apps-india-unsafe-unsecured/, August 2015.

18. Richard Handford. Mobile World Live - Survey finds security holes in mobile money apps. http://www.mobileworldlive.com/money/news-money/survey-finds-security-holes-in-mobile-money-apps/#.Vc27Y-QTmSQ.twitter, August 2015.

19. JENNIFER VALENTINO-DEVRIES. Wall Street Journal - Researchers Find Security Flaws in Developing-World Money Apps. http://blogs.wsj.com/digits/2015/08/11/researchers-find-security-flaws-in-developing-world-money-apps/, August 2015.

20. Jonathon Cheng. Wall Street Journal - Samsung Phone Studied for Possible Security Gap. http://online.wsj.com/news/articles/SB10001424052702304244904579276191788427198, December 2013.

21. N. V. The Economist - The Threat in the Pocket. http://www.economist.com/blogs/babbage/2013/10/difference-engine-0?fsrc=scn/fb/wl/bl/thethreatinthepocket, October 2013.

22. Antone Gonsalves. ComputerWorld - Let's Dump Anti-Virus and Move On:. http://blogs.computerworld.com/mobile-security/22969/lets-dump-av-and-move, October 2013.

23. Mathew J. Schwartz. InformationWeek - Google: Don't Fear Android Malware. http://www.informationweek.com/security/mobile/google-dont-fear-android-malware/240162399, October 2013.

24. Kirsten Doyle. ITWeb - Android Threat Exaggerated, or is it? http://www.itweb.co.za/index.php?option=com_content&view=article&id=68055, October 2013.

25. Danielle Walker. SC Magazine - Mobile malware prevalence expands, but privacy-abusing apps should be top of mind. http://www.scmagazine.com/mobile-malware-prevalence-expands-but-privacy-abusing-apps-should-be-top-of-mind/article/300597/, June 2013.

26. Jim Burress. WABE NPR - Mobile Web Browsers Full of Security Risks, Tech Professor Finds. http://wabe.org/post/mobile-web-browsers-full-security-risks-tech-professor-finds, December 2012.

27. Mark Huffman. Consumer Affairs - Georgia Tech: mobile browsers fail safety test. http://www.consumeraffairs.com/news/georgia-tech-mobile-browsers-fail-safety-test-120612.html, December 2012.

28. Matthew J. Schwartz. Information Week - Blame Screen Size: Mobile Browsers Flunk Security Tests. http://www.informationweek.com/security/mobile/blame-screen-size-mobile-browsers-flunk/240143999, December 2012.

29. Jon Gold. Network World - Ga. Tech researchers: Mobile Browsers need better HTTPS indicators. http://www.networkworld.com/news/2012/120512-mobile-browsers-264846.html, December 2012.

30. United Press International. Study: Most mobile Web browsers unsafe. http://www.upi.com/Science_News/Technology/2012/12/05/Study-Most-mobile-Web-browsers-unsafe/UPI-73431354743353/#ixzz2EGtQsuLd, December 2012.

31. Suzanne Choney. Mobile browser woes can fool even experts: report . http://www.nbcnews.com/technology/mobile-browser-woes-can-fool-even-experts-report-1C7451203, December 2012.

32. Meghan Kelly. VentureBeat - 3 hot security startups to watch. http://venturebeat.com/2012/02/27/3-security-startups-to-watch-at-the-2012-rsa-conference/, February 2012.

33. Jacob Goodwin. Government Security News - RSA 2012 – Pindrop Security can distinguish a fraudulent phone call from a real one. http://www.gsnmagazine.com/node/25721?c=communications, February 2012.

34. Matt Liebowitz. Phone hack logs keystrokes from nearby computers. MSNBC.com - http://www.msnbc.msn.com/id/44993238/ns/technology_and_science-security/#.TqU5MNSjPh4, October 2011.

35. Jacob Aron. iPhone keylogger can snoop on desktop typing. New Scientist - http://www.newscientist.com/article/dn21059-iphone-keylogger-can-snoop-on-desktop-typing.html, October 2011.

36. iPhone Keylogger Can Snoop on Desktop Typing. Slashdot - http://mobile.slashdot.org/story/11/10/18/2346222/iphone-keylogger-can-snoop-on-desktop-typing, October 2011.

37. Robert Lemos. Smart Phones Could Hear Your Password. Technology Review - http://www.technologyreview.com/computing/38913/?p1=A2, October 2011.

38. Kevin McCaney. Bad vibrations: How smart phones could steal PC passwords. Government Computer News - http://gcn.com/articles/2011/10/18/smart-phone-sensors-steal-keystrokes.aspx, October 2011.

39. PhysOrg. Turning iPhone into spiPhone: Smartphones' accelerometer can track strokes on nearby keyboards. PhysOrg.com - http://www.physorg.com/news/2011-10-iphone-spiphone-smartphones-accelerometer-track.html, October 2011.

40. Brid-Aine Parnell. Securo-boffins call for 'self-aware' defensive technologies. The Register - http://www.theregister.co.uk/2011/09/14/self_aware_cyber_security_technologies_should_be_a_top_priority/, September 2011.

41. Clay Dillow. 'PinDr0p' Tech Uses Unique Noise Fingerprints to Trace Calls. Popular Science - http://www.popsci.com/technology/article/2010-10/pindr0p-tech-tags-phone-calls-unique-fingerprints-trace-call-paths-across-networks, October 2010.

42. Lewis Page. Voice-routing call fingerprint system fights vishing. The Register - http://www.theregister.co.uk/2010/10/06/voice_fingerprints, October 2010.

43. Science Daily. Voice Phishing: System to Trace Telephone Call Paths Across Multiple Networks Developed. http://www.sciencedaily.com/releases/2010/10/101005121820.htm, October 2010.

44. Brian Kalish. To Text or Not to Text During Emergencies. NextGov.com - http://www.nextgov.com/nextgov/ng_20100914_5986.php?oref=topnews, September 2010.

45. Ki Mae Heussner. 'Operation Chokehold': Fake Steve Jobs Rallies iPhone Users to Cripple AT&T Network. ABC News - http://abcnews.go.com/Technology/GadgetGuide/fake-steve-jobs-rallies- iphone-users-cripple-att/story?id=9355447, December 2009.

46. Bob Brown. Researchers Set Their Sights on iPhones, Mobile Malware. PC World Magazine - ihttp://www.pcworld.com/article/182005/iphone_worms_mobile_malware.html?tk=rss, November 2009.

47. MacGregor Campbell. Botnets show their disruptive potential. New Scientist Magazine - http://www.newscientist.com/article/mg20427347.000-mobile-botnets-show-their-disruptive-potential.html, November 2009.

48. Angela Moscaritolo. Remote repair for infected phones in development. SC Magazine - http://www.scmagazineus.com/remote-repair-for-infected-phones-in- development/article/157504/, November 2009.

49. Bob Brown. iPhone worms, other smartphone malware in researchers' sights. Network World - http://www.networkworld.com/news/2009/111109-smartphone-security-georgia -tech.html?hpg1=bn, November 2009.

50. Urvaksh Karkaria. GT researchers work to secure cellphones. Atlanta Business Chronicle - http://atlanta.bizjournals.com/atlanta/blog/atlantech/2009/11/cellphone.html, November 2009.

51. Making Carriers Shoulder Smartphone Security. http://mobile.slashdot.org/story/09/11/11 /2318247/Making-Carriers-Shoulder-Smartphone-Security?art_pos=31, November 2009.

52. Ben Meyer. Georgia Tech to Lead Fight Against Cell Phone Hackers. NBC 11 Atlanta - http://www.11alive.com/news/local/story.aspx?storyid=132505&catid=3, July 2009.

53. Illena Armstrong. Safeguarding your mobile networks. SC Magazine - http://www.scmagazineus.com/Safeguarding-your-mobile-networks/article/138289/, June 2009.

54. Kelli B. Grant. Four Free Cellphone Apps to Help Manage Your Money. SmartMoney Magazine - http://www.smartmoney.com/Spending/Deals/4-Great-Free-Finance-Apps-for-Cellphones/, June 2009.

55. Amanda Hoffstrom. Technology's limitations in alerting campus danger. UWire Magazine - http://www.uwire.com/Article.aspx?id=3738798, February 2009.

56. Laura Sydell. Compromise Allows Obama To Keep BlackBerry. National Public Radio - http://www.npr.org/templates/story/story.php?storyId=99790788, January 2009.

57. Dennis Carter. Questions abound as emergency alert flops Virginia Tech's text-message alert system failed when the sound of gunfire was heard on campus; officials scramble to understand why. eSchool News - http://www.eschoolnews.com/iphone/top-story/index.cfm?i=56122#_56122, November 2008.

58. Jessica Bauer. Study: Text alerts may fail in real emergency. Diamondback Online - http://media.www.diamondbackonline.com/media/storage/paper873/news/2008/10/14/News/Study.Text.Alerts.May.Fail.In.Real.Emergency-3485509.shtml, October 2008.

59. Associated Press. Hackers Expected To Start Targeting Cell Phones. http://cbs5.com/watercooler/Cell.Phones.Hackers.2.840909.html, 2008.

60. Associated Press. College alert systems unreliable, study says. http://www.ajc.com/search/content/metro/stories/2008/09/25/college_campus_alerts.html, 2008.

61. Lee Shearer. Study: Campus alerts unreliable. Athens Banner Herald http://www.onlineathens.com/stories/092508/uga_336494829.shtml, 2008.

62. Bill Ray. 3G Americas warns against text warning systems. The Register - http://www.theregister.co.uk/2008/09/18/emergency_text/, 2008.

63. 3G Americas. 3G Americas Highlights New Research Report on Use of Cellular Text Messaging for Emergency Alert Services. 3G Americas http://www.3gamericas.org/English/news_room/DisplayPressRelease.cfm?id=3400&s=ENG, 2008.

64. Evan Koblentz. Web Exclusive: From Messaging to Management Duty. Wireless Week - http://www.wirelessweek.com/Messaging-to-Management-Duty.aspx, 2008.

65. Christopher Beam. How Do You Intercept a Text Message? Turn your cell phone into a spy gadget. Slate Magazine http://www.slate.com/id/2161402/, 2007.

66. Jamming Cellphones with Text Messages. Slashdot http://it.slashdot.org/it/05/10/05/1839217.shtml?tid=215&tid=172, 2005.

67. Cell phone networks at risk? CNN http://money.cnn.com/2005/10/05/technology/hacker_cellphones/, 2005.

68. John Schwartz. Text Hackers Could Jam Cellphones, a Paper Says. The New York Times http://www.nytimes.com/2005/10/05/technology/05phone.html?ex=1286164800&en=d917b9cd43dfaa31&ei=5090&partner=rssuserland&emc=rss, 2005.