

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM SERVICES LLC,
Petitioner

v.

HEADWATER RESEARCH LLC,
Patent Owner.

IPR2027-00015

U.S. Patent No. 8,832,777

DECLARATION OF CHARLES CREUSERE, PH.D.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further, 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 Title 18 of the United States Code.

Date:

10/22/2026

By:



Charles Creusere, Ph.D.

LISTING OF CHALLENGED CLAIMS.....	XII
I. INTRODUCTION	1
A. Educational Background	1
B. Materials Considered.....	6
II. OVERVIEW AND LEGAL STANDARDS	7
A. Person of Ordinary Skill in the Art	8
B. Obviousness.....	8
C. Claim Construction	13
III. OVERVIEW OF THE '777 PATENT	14
A. Summary of the '777 Patent.....	14
B. Field of Endeavor	18
C. Problem Allegedly Solved by the Inventors of the '777 Patent	19
D. File History of the '777 Patent	22
IV. LEVEL OF A PERSON OF ORDINARY SKILL	27
V. OVERVIEW OF THE TECHNOLOGY	28
A. Device Agents	29
B. Device Credentials and Network Admission	29
C. Service Processor Authentication.....	30
D. Network Records of Device Usage	30
VI. SUMMARY OF THE PRIOR ART REFERENCES	31
A. Overview of Wright.....	31
B. Overview of Bittmann.....	45
C. Overview of Durham.....	53
VII. SUMMARY OF UNPATENTABILITY	59
VIII. GROUNDS	60
A. GROUND 1A: CLAIMS 1–7, 10, 11, 13, AND 14 ARE OBVIOUS OVER WRIGHT	60

1.	Claim 1	60
a.	1[pre]: “A method performed by a network system, the method comprising:”	60
b.	1[a]: “receiving a device credential from an end-user device communicatively coupled to the network system over a wireless access network;”	62
c.	1[b]: “obtaining a service policy based at least in part on the device credential, the service policy being applicable to the end-user device;”	77
d.	1[c][i]: “obtaining service processor authentication information associated with a service processor on the end-user device,”	86
e.	1[c][ii]: “the service processor comprising one or more device agents at least assisting to implement the service policy; and”	90
f.	1[d]: “using the service processor authentication information, facilitating execution of an end-user device service processor authentication procedure.”	93
2.	Claim 2: “The method recited in claim 1, wherein the network system comprises an authentication, authorization, and accounting (AAA) server, a home location register (HLR), a policy charging and rules function (PCRF), an access network authentication system, an admission system, a log-in system, or a combination of these.”	95
3.	Claim 3: “The method recited in claim 1, wherein the network system comprises a service controller, a gateway, a router, a gateway general packet radio service (GPRS) support node (GGSN), a serving GPRS support node (SGSN), a proxy, a charging element, a notification trigger element, or a combination of these.”	97

4. Claim 4: “The method recited in claim 1, wherein facilitating execution of the end-user device service processor authentication procedure comprises verifying that the service processor is present and configured in an expected manner.”98
5. Claim 5: “The method recited in claim 1, further comprising executing a first portion of the service policy when the end-user device service processor authentication procedure is successful.”100
6. Claim 6: “The method recited in claim 5, further comprising executing a second portion of the service policy when the end-user device service processor authentication procedure is unsuccessful.”102
7. Claim 7: “The method recited in claim 1, wherein the device credential comprises information from a subscriber identity module (SIM) card, information from a soft SIM, information from a universal SIM, an international mobile subscriber identity (IMSI), a wireless modem identifier, a phone number, an international mobile equipment identity (IMEI), a mobile equipment identifier (MEID), a media access control (MAC) address, an Internet protocol (IP) address, a device identifier, an encryption key, or a combination of these.”104
8. Claim 10: “The method recited in claim 1, wherein the service processor comprises a client configured to implement at least a portion of the service policy.”106
9. Claim 11: “The method recited in claim 1, wherein the service policy comprises a control policy, a service usage limit, an accounting policy, a billing policy, a notification policy, or a combination of these.”107
10. Claim 13: “The method recited in claim 5, wherein executing the first portion of the service policy

	comprises sending the first portion of the service policy to the service processor.”	107
11.	Claim 14: “The method recited in claim 6, wherein the second portion of the service policy differs from the first portion of the service policy in a network access privilege, an allowable network destination, a service usage accounting or billing policy, a service usage rate, a service speed or quality, an allowed access network, a user notification, a roaming policy or permission, an effect on a device operation, or a combination of these.”	108
B.	GROUND 1B: CLAIMS 15–17, 19–22, 28, 29, 31, AND 32 ARE OBVIOUS OVER WRIGHT AND BITTMANN	109
1.	Bittmann’s Teachings	109
2.	Motivation to Combine Wright and Bittmann.....	110
3.	Claim 15.....	124
a.	15[a]: “The method recited in claim 1, further comprising: receiving a data usage report associated with the end-user device from a good customer feedback source; and”	124
b.	15[b]: “based on the data usage report, adjusting a user account, the user account being associated with the end-user device.”	130
4.	Claim 16.....	132
a.	16[a]: “The method recited in claim 1, further comprising: receiving a device data usage report from the end-user device”	132
b.	16[b]: “obtaining a trusted data usage report associated with the end-user device; and”	134
c.	16[c]: “reconciling the device data usage report and the trusted data usage report.”	137
5.	Claim 17: “The method recited in claim 16, wherein obtaining the trusted data usage report comprises obtaining the trusted data usage report from a network element.”	139

6. Claim 19: “The method recited in claim 16, wherein obtaining the trusted data usage report comprises obtaining the trusted data usage report from a good customer feedback source.”140
7. Claim 20: “The method recited in claim 19, wherein the good customer feedback source comprises a website or server that tracks a number of visits by the end-user device, a number of transactions associated with the end-user device, an amount of business generated in association with the end-user device, an amount of data communicated by or to the end-user device, a measure of an interaction by the end-user device with the website or server, or a combination of these.”141
8. Claim 21: “The method recited in claim 16, wherein the trusted data usage report comprises a flow data record.”142
9. Claim 22: “The method recited in claim 16, further comprising adjusting a user account associated with the end-user device based on reconciling the device data usage report and the trusted data usage report.”145
10. Claim 28: “The method recited in claim 16, wherein the device data usage report indicates usage associated with a classification, the classification being one or more device applications, one or more network destinations, one or more network types, one or more sponsored services, one or more background activities, one or more operating system (OS) programs, one or more OS libraries, one or more OS functions, or a combination of these.”147
11. Claim 29: “The method recited in claim 16, wherein the trusted data usage report indicates usage associated with a classification, the classification being one or more device applications, one or more network destinations, one or more network types,

one or more sponsored services, one or more background activities, one or more operating system (OS) programs, one or more OS libraries, one or more OS functions, or a combination of these.”	148
12. Claim 31: “The method recited in claim 16, wherein the device data usage report comprises information identifying an application, an operating system (OS) function, an OS program, an OS library, a network resource, a web site, a network source, a network destination, a domain, a uniform resource locator (URL), an IP address, a port, a protocol, a socket tuple, a server, a network route, an access point name (APN), a gateway, a proxy, a content source, a sub-network, a quality-of-service (QoS) level, a content type, a background activity, or a combination of these.”	150
13. Claim 32: “The method recited in claim 16, wherein the trusted data usage report comprises information identifying an application, an operating system (OS) function, an OS program, an OS library, a network resource, a web site, a network source, a network destination, a domain, a uniform resource locator (URL), an IP address, a port, a protocol, a socket tuple, a server, a network route, an access point name (APN), a gateway, a proxy, a content source, a sub-network, a quality-of-service (QoS) level, a content type, a background activity, or a combination of these.”	151
C. GROUND 1C: CLAIMS 8, 9, 12, AND 23–25 ARE OBVIOUS OVER WRIGHT AND DURHAM.....	152
1. Durham’s Teachings	152
2. Motivation to Combine Wright and Durham.....	154
3. Claim 8	161
a. 8[a]: “The method recited in claim 1, wherein the device credential is a first device credential, and wherein the end-user device is	

- a first end-user device, and wherein the service policy is a first access network control policy, and further comprising:”161
- b. 8[b]: “associating the first device credential with the first end-user device;”163
- c. 8[c]: “determining whether a second device credential, associated with an authenticating end-user device, is the first device credential;”167
- d. 8[d]: “determining whether the authenticating end-user device is the first end-user device;”168
- e. 8[e]: “enforcing the first access network control policy when the second device credential is the first device credential and the authenticating end-user device is the first end-user device; and”175
- f. 8[f]: “enforcing a second access network control policy when the second device credential is not the first device credential or the authenticating end-user device is not the first end-user device.”177
- 4. Claim 9: “The method recited in claim 8, wherein the second access network control policy differs from the first access network control policy in a network access privilege, an allowable network destination, a service usage accounting or billing policy, a service usage rate, a service speed or quality, an allowed access network, a user notification, a roaming policy or permission, an operation of the end-user device, or a combination of these.”181
- 5. Claim 12: “The method recited in claim 8, wherein enforcing the second access network control policy comprises suspending or blocking network access by the authenticating end-user device.”183

6.	Claim 23	184
a.	23[a]: “The method recited in claim 1, further comprising: detecting a service usage indication associated with the end-user device;”	184
b.	23[b]: “detecting a specified condition; and”	186
c.	23[c]: “declaring a service processor integrity violation based on detecting the specified condition.”	187
7.	Claim 24: “The method recited in claim 23, wherein the specified condition comprises that the service processor does not communicate with the network system within a predetermined amount of time after the network system detects the service usage indication.”	190
8.	Claim 25: “The method recited in claim 23, wherein the specified condition comprises that the network system is unable to complete the end-user device service processor authentication procedure.”	193
D.	GROUND 1D: CLAIMS 16, 18, 23, 26, 27, 30, AND 33 ARE OBVIOUS OVER WRIGHT, BITTMANN, AND DURHAM	195
1.	Motivation to Combine Wright, Bittmann, and Durham.....	195
2.	Claims 16/18/33	205
a.	Claim 16.....	205
(i)	16[a]: “The method recited in claim 1, further comprising: receiving a device data usage report from the end-user device;”	205
(ii)	16[b]: “obtaining a trusted data usage report associated with the end-user device; and”	206

(iii)	16[c]: “reconciling the device data usage report and the trusted data usage report.”	210
b.	Claim 18: “The method recited in claim 16, wherein obtaining the trusted data usage report comprises obtaining the trusted data usage report from a software or firmware agent in a secure execution environment on the end-user device.”	211
c.	Claim 33: “The method recited in claim 18, wherein the software or firmware agent is part of the service processor.”	213
3.	Claims 23/26/27/30	214
a.	Claim 23	214
(i)	23[a]: “The method recited in claim 1, further comprising: detecting a service usage indication associated with the end-user device”	214
(ii)	23[b]: “detecting a specified condition; and”	215
(iii)	23[c]: “declaring a service processor integrity violation based on detecting the specified condition.”	218
b.	Claim 26: “The method recited in claim 23, wherein the service usage indication is a first service usage indication, and further comprising receiving a second service usage indication from the service processor, and wherein the specified condition comprises that the second service usage indication and the first service usage indication are inconsistent.”	221
c.	Claim 27: “The method recited in claim 23, wherein the service usage indication is a first service usage indication, and further comprising receiving a second service usage	

	indication from the service processor, and wherein the specified condition comprises that either the first service usage indication or the second service usage indication is inconsistent with the service policy.”	223
d.	Claim 30: “The method recited in claim 23, wherein the service usage indication comprises a start accounting message, a stop accounting message, a charging data record, a flow data record, a service usage measure, or a combination of these.”	225
IX.	SUMMARY AND ADDITIONAL REMARKS.....	226

LISTING OF CHALLENGED CLAIMS

Reference	Limitation
Claim 1	
1[pre]	A method performed by a network system, the method comprising:
1[a]	receiving a device credential from an end-user device communicatively coupled to the network system over a wireless access network;
1[b]	obtaining a service policy based at least in part on the device credential, the service policy being applicable to the end-user device;
1[c][i]	obtaining service processor authentication information associated with a service processor on the end-user device,
1[c][ii]	the service processor comprising one or more device agents at least assisting to implement the service policy; and
1[d]	using the service processor authentication information, facilitating execution of an end-user device service processor authentication procedure.
Claim 2	
2	The method recited in claim 1, wherein the network system comprises an authentication, authorization, and accounting (AAA) server, a home location register (HLR), a policy charging and rules function (PCRF), an access network authentication system, an admission system, a log-in system, or a combination of these.
Claim 3	
3	The method recited in claim 1, wherein the network system comprises a service controller, a gateway, a router, a gateway general packet radio service (GPRS) support node (GGSN), a serving GPRS support node (SGSN), a proxy, a charging element, a notification trigger element, or a combination of these.

Reference	Limitation
Claim 4	
4	The method recited in claim 1, wherein facilitating execution of the end-user device service processor authentication procedure comprises verifying that the service processor is present and configured in an expected manner.
Claim 5	
5	The method recited in claim 1, further comprising executing a first portion of the service policy when the end-user device service processor authentication procedure is successful.
Claim 6	
6	The method recited in claim 5, further comprising executing a second portion of the service policy when the end-user device service processor authentication procedure is unsuccessful.
Claim 7	
7	The method recited in claim 1, wherein the device credential comprises information from a subscriber identity module (SIM) card, information from a soft SIM, information from a universal SIM, an international mobile subscriber identity (IMSI), a wireless modem identifier, a phone number, an international mobile equipment identity (IMEI), a mobile equipment identifier (MEID), a media access control (MAC) address, an Internet protocol (IP) address, a device identifier, an encryption key, or a combination of these.
Claim 8	
8[a]	The method recited in claim 1, wherein the device credential is a first device credential, and wherein the end-user device is a first end-user device, and wherein the service policy is a first access network control policy, and further comprising:

Reference	Limitation
8[b]	associating the ¹ first device credential with the first end-user device;
8[c]	determining whether a second device credential, associated with an authenticating end-user device, is the first device credential;
8[d]	determining whether the authenticating end-user device is the first end-user device;
8[e]	enforcing the first access network control policy when the second device credential is the first device credential and the authenticating end-user device is the first end-user device; and
8[f]	enforcing a second access network control policy when the second device credential is not the first device credential or the authenticating end-user device is not the first end-user device.
Claim 9	
9	The method recited in claim 8, wherein the second access network control policy differs from the first access network control policy in a network access privilege, an allowable network destination, a service usage accounting or billing policy, a service usage rate, a service speed or quality, an allowed access network, a user notification, a roaming policy or permission, an operation of the end-user device, or a combination of these.
Claim 10	
10	The method recited in claim 1, wherein the service processor comprises a client configured to implement at least a portion of the service policy.

¹ March 13, 2018 certificate corrected claim language from “associating first” to “associating the first.” Ex-1001.

Reference	Limitation
Claim 11	
11	The method recited in claim 1, wherein the service policy comprises a control policy, a service usage limit, an accounting policy, a billing policy, a notification policy, or a combination of these.
Claim 12	
12	The method recited in claim 8, wherein enforcing the second access network control policy comprises suspending or blocking network access by the authenticating end-user device.
Claim 13	
13	The method recited in claim 5, wherein executing the first portion of the service policy comprises sending the first portion of the service policy to the service processor.
Claim 14	
14	The method recited in claim 6, wherein the second portion of the service policy differs from the first portion of the service policy in a network access privilege, an allowable network destination, a service usage accounting or billing policy, a service usage rate, a service speed or quality, an allowed access network, a user notification, a roaming policy or permission, an effect on a device operation, or a combination of these.
Claim 15	
15[a]	The method recited in claim 1, further comprising: receiving a data usage report associated with the end-user device from a good customer feedback source; and
15[b]	based on the data usage report, adjusting a user account, the user account being associated with the end-user device.

Reference	Limitation
Claim 16	
16[a]	The method recited in claim 1, further comprising: receiving a device data usage report from the end-user device;
16[b]	obtaining a trusted data usage report associated with the end-user device; and
16[c]	reconciling the device data usage report and the trusted data usage report.
Claim 17	
17	The method recited in claim 16, wherein obtaining the trusted data usage report comprises obtaining the trusted data usage report from a network element.
Claim 18	
18	The method recited in claim 16, wherein obtaining the trusted data usage report comprises obtaining the trusted data usage report from a software or firmware agent in a secure execution environment on the end-user device.
Claim 19	
19	The method recited in claim 16, wherein obtaining the trusted data usage report comprises obtaining the trusted data usage report from ² a good customer feedback source.
Claim 20	
20	The method recited in claim 19, wherein the good customer feedback source comprises a website or server that tracks a number of visits by

² March 13, 2018 certificate corrected claim language from “report is from” to “report from.” Ex-1001.

Reference	Limitation
	the end-user device, a number of transactions associated with the end-user device, an amount of business generated in association with the end-user device, an amount of data communicated by or to the end-user device, a measure of an interaction by the end-user device with the website or server, or a combination of these.
Claim 21	
21	The method recited in claim 16, wherein the trusted data usage report comprises a flow data record.
Claim 22	
22	The method recited in claim 16, further comprising adjusting a user account associated with the end-user device based on reconciling the device data usage report and the trusted data usage report.
Claim 23	
23[a]	The method recited in claim 1, further comprising: detecting a service usage indication associated with the end-user device;
23[b]	detecting a specified condition; and
23[c]	declaring a service processor integrity violation based on detecting the specified condition.
Claim 24	
24	The method recited in claim 23, wherein the specified condition comprises that the service processor does not communicate with the network system within a predetermined amount of time after the network system detects the service usage indication.
Claim 25	
25	The method recited in claim 23, wherein the specified condition comprises that the network system is unable to complete the end-user device service processor authentication procedure.

Reference	Limitation
Claim 26	
26	The method recited in claim 23, wherein the service usage indication is a first service usage indication, and further comprising receiving a second service usage indication from the service processor, and wherein the specified condition comprises that the second service usage indication and the first service usage indication are inconsistent.
Claim 27	
27	The method recited in claim 23, wherein the service usage indication is a first service usage indication, and further comprising receiving a second service usage indication from the service processor, and wherein the specified condition comprises that either the first service usage indication or the second service usage indication is inconsistent with the service policy.
Claim 28	
28	The method recited in claim 16, wherein the device data usage report indicates usage associated with a classification, the classification being one or more device applications, one or more network destinations, one or more network types, one or more sponsored services, one or more background activities, one or more operating system (OS) programs, one or more OS libraries, one or more OS functions, or a combination of these.
Claim 29	
29	The method recited in claim 16, wherein the trusted data usage report indicates usage associated with a classification, the classification being one or more device applications, one or more network destinations, one or more network types, one or more sponsored services, one or more background activities, one or more operating system (OS) programs, one or more OS libraries, one or more OS functions, or a combination of these.

Reference	Limitation
Claim 30	
30	The method recited in claim 23, wherein the service usage indication comprises a start accounting message, a stop accounting message, a charging data record, a flow data record, a service usage measure, or a combination of these.
Claim 31	
31	The method recited in claim 16, wherein the device data usage report comprises information identifying an application, an operating system (OS) function, an OS program, an OS library, a network resource, a web site, a network source, a network destination, a domain, a uniform resource locator (URL), an IP address, a port, a protocol, a socket tuple, a server, a network route, an access point name (APN), a gateway, a proxy, a content source, a sub-network, a quality-of-service (QoS) level, a content type, a background activity, or a combination of these.
Claim 32	
32	The method recited in claim 16, wherein the trusted data usage report comprises information identifying an application, an operating system (OS) function, an OS program, an OS library, a network resource, a web site, a network source, a network destination, a domain, a uniform resource locator (URL), an IP address, a port, a protocol, a socket tuple, a server, a network route, an access point name (APN), a gateway, a proxy, a content source, a sub-network, a quality-of-service (QoS) level, a content type, a background activity, or a combination of these.
Claim 33	
33	The method recited in claim 18, wherein the software or firmware agent is part of the service processor.

I, Dr. Charles D. Creusere, Ph. D., do hereby declare as follows:

I. INTRODUCTION

1. I have been retained by counsel for Petitioner as a technical expert in the above-captioned case. Specifically, I have been asked to render certain opinions regarding the IPR petition with respect to U.S. Patent No. 8,832,777 (the “’777 Patent”). I understand that the Challenged Claims are 1-33, and my opinions are limited to those claims. A true and correct copy of my Curriculum Vitae, which provides further details about my background and experience, is appended to this Declaration.

A. Educational Background

2. I received a Bachelor of Science degree in Electrical and Computer Engineering from the University of California at Davis in 1985. I received a Master of Science degree in Electrical and Computer Engineering from the University of California at Santa Barbara in 1990, and I received my Ph.D. in Electrical and Computer Engineering, also from the University of California at Santa Barbara, in 1993.

3. I am currently a Full Professor in the Klipsch School of Electrical & Computer Engineering at New Mexico State University, and I held the International Foundation for Telemetry from 2008 to 2010 and the Frank Carden Endowed Chair in Telemetry and Telecommunications from 2010 to 2024. I was an

Assistant Professor at New Mexico State from January 2000 until I became an Associate Professor in 2004. I have been a Full Professor since August 2010. My research and teaching at New Mexico State have focused on digital signal processing and communications.

4. I have extensive experience in the technical areas of the '777 Patent, including more than 30 years of experience with digital systems, data communications, and communications theory.

5. After receiving my B.S. from U.C. Davis in 1985, I went to work for the Naval Weapons Center, China Lake as a civilian Department of Defense (DoD) employee, where I was the lead designer for the guidance electronics of the Laser Guided Training Round. My work on this project included analog and digital circuit design, embedded software design, and front-end systems integration and testing. I was awarded a fellowship from the DoD to pursue graduate degrees at the University of California, Santa Barbara. During my Masters program at UCSB, I pursued a dual communications/digital signal processing coursework specialization. Several classes I took have relevance to the patent in suit, but ECE243 (Digital Communications) is particularly pertinent here given that digital communication is a critical part of any wireless network system.

6. While the focus of my Ph.D. research was on multirate filter banks and signal compression, I nonetheless followed the communications field closely during

the six years after receiving my Ph.D. when I was a civilian working for the department of defense at the Naval Air Warfare Center (now NAVAIR), China Lake, from 1993 through 1999. I was a Researcher and Team Leader at NAVAIR. As part of my work there, I developed efficient video compression and decompression algorithms for use with wireless communication channels. This work was presented at two conferences (the 1996 SPIE Conference and the 1996 Data Compression Conference) and it was published in the October 1997 issue of IEEE Transactions on Image Processing. In addition, the complete real-time internet streaming system was documented in NAWCWD Technical Paper, TP 8442.

7. Since joining the faculty of New Mexico State University in 2000, I have taught numerous classes at both the graduate and undergraduate levels. Many of the classes I have taught are relevant to the '777 Patent. These include EE497 (Digital Communications), EE573 (Signal Compression), and EE586 (Information Theory). Particularly relevant to the patent-in-suit is EE585 (Telemetry Systems) which I began teaching regularly in 2009 shortly after being selected for the IFT Professorship. The focus of this class was the wireless transmission of data and amongst the topics covered was iNet Telemetric Networks, a packet-based telemetry standard designed to facilitate internet-like connectivity of devices over military radios. Many of the problems that the patent-in-suit claims to address (e.g.,

the security of data communicated over wireless networks) are also significant issues for the military-centric iNet network.

8. I have also done research in areas related to communications systems during my tenure at NMSU. The first was for a 2006 DARPA-funded project titled “ADAM Project” that we executed in collaboration with partners at Los Alamos National Laboratories. As part of this project, we implemented in the Mathwork’s Simulink environment a complete simulation of a satellite-to-ground communications system to model the effects of specific compressive-sensing-related processing techniques on signal recovery. Also in 2006, I directed an ARO-funded research project dealing with distributed compression of audio and video collected from a set of distributed sensing nodes. As part of this effort, we created the Distributed Coding Testbed consisting of four sensor nodes that communicated with one another via WiFi and Bluetooth using standard TCP/IP protocols in order to jointly coordinate the compression of sensed audio and video to achieve maximum overall efficiency. From 2011-2014, I was a co-PI on a NASA-funded project entitled “Proximity Operations for Near Earth Asteroid Exploration.” My portion of the project dealt with the communications between the spacecraft orbiting the asteroid and multiple probes on its surface, specifically how one might use the communications waveform along with orbital prediction to estimate the relative location of the various parties.

9. More recently, I worked with the Air Force Research Lab to design a software-defined radio capable of fingerprinting satellites from the electronic signature of their radio. I have also done research involving communications at the network level, including assisting Dr. Justin MacDonald, a professor at NMSU in the Department of Psychology, with his NIH-funded project on Gaze-Directed Hearing Assistance/Augmentation, as well as leading an internal project studying the feedback dynamics of brain-computer interfaces in a distributed processing-computing environment. This project involved real-time asynchronous network communications between multiple programs (agents) that were running simultaneously on multiple networked computer systems.

10. From 1990 through 1993, I worked as a Research Assistant in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. In this position, I worked on subband coding (compression) and multirate filter bank theory. I also implemented real-time filter banks on a digital signal processor. In the summer of 1992, I worked at AT&T Bell Labs, where I developed and simulated new methods of extremely low-bit-rate video coding for video telephone applications.

11. A listing of the cases (including trials before the Patent Trial and Appeal Board) in which I have testified is found following my CV in the attached Appendix A.

B. Materials Considered

12. I have relied upon my education, knowledge, and experience with data communications, as well as the other materials as discussed in this declaration in forming my opinions.

13. In developing my opinions, I have considered the following materials:

No.	Exhibit
1001	U.S. Patent No. 8,832,777 (“’777 patent” or “’777Pat”)
1003	File History of U.S. Patent No. 8,832,777 (copies of prior art references omitted)
1004	U.S. Patent Pub. No. 2004/0123153 to Wright et al. (“Wright”)
1005	U.S. Patent Pub. No. 2004/0148237 to Bittmann et al. (“Bittmann”)
1006	U.S. Patent Pub. No. 2007/0006282 to Durham et al. (“Durham”)
1007	U.S. Patent No. 7,353,533 to Wright et al. (“the ’533 patent”)
1008	U.S. Patent Pub. No. 2008/0109679 to Wright et al. (“the ’679 publication”)
1009	U.S. Patent No. 7,478,420 to Wright et al. (“the ’420 patent”)
1010	IPR2024-00809, Final Written Decision
1011	U.S. Patent Pub. No. 2006/0104264 to Cho (“Cho”)

14. I have considered these materials from the viewpoint of a POSITA as of the priority date of the ’777 Patent. For the purposes of this declaration, I have been asked to assume that the priority date of the ’777 Patent is January 28, 2009. I

note that my opinions provided in this Declaration are made from the perspective of a POSITA as of this priority date of the '777 Patent unless expressly stated otherwise. To the extent that I use any verb tense in this Declaration that is present tense (e.g., “a POSITA would understand” instead of “a POSITA would have understood”), such verb tense should be understood to be my opinion as of the '777 Patent’s priority date (again, unless expressly stated otherwise). I merely use the present verb tense for ease of reading.

II. OVERVIEW AND LEGAL STANDARDS

15. In formulating my opinions, I have been instructed to apply certain legal standards. I am not a lawyer. I do not offer any testimony regarding what the law is. Instead, the following sections summarize the law as I have been instructed to apply it in formulating and rendering my opinions found later in this declaration. I understand that, in an *inter partes* review (“IPR”) proceeding, patent claims may be deemed unpatentable if it is shown that they are anticipated or rendered obvious in view of the prior art. I understand that prior art in an IPR review is limited to patents or printed publications that predate the priority date of the patent at issue. I understand that questions of claim clarity (definiteness) and enablement cannot be considered as a ground for considering the patentability of a claim in these proceedings.

A. Person of Ordinary Skill in the Art

16. I understand that the '777 Patent, the record of the proceedings at the Patent Office (which I understand is called the “File History” or “Prosecution History”), and the teachings of the prior art are evaluated from the perspective of a person of ordinary skill in the art (“POSITA”). I understand that the factors considered in determining the ordinary level of skill in the art may include: (i) the levels of education of the inventor; (ii) the types of problems encountered in the art; (iii) prior-art solutions to those problems; (iv) the rapidity with which innovations are made; (v) the sophistication of the technology; and (vi) the educational level of persons working in the field.

17. I understand that a person of ordinary skill in the art is not a specific real individual, but rather a hypothetical individual having the qualities reflected by the factors above. The hypothetical person is presumed to have the same level of skill as the typical practitioner of the art and is presumed to have knowledge of all prior art in the relevant field. I understand that the inventor’s actual knowledge or lack of knowledge of prior art reference is irrelevant to the obviousness determination.

B. Obviousness

18. I understand that a claim may be invalid under 35 U.S.C. § 103(a) if the subject matter described by the claim as a whole would have been “obvious” to a

POSITA in view of a single or combination of prior-art references at the time the claimed invention was made. I further understand that a POSITA is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent-in-suit and all analogous prior art. I understand that obviousness in an IPR review proceeding is evaluated using a preponderance of the evidence standard, which means that the claims must be more likely obvious than nonobvious.

19. I also understand that an obviousness determination includes the consideration of various factors including: (1) the scope and content of the prior art, (2) the differences between the prior art and the claim at issue, and (3) the level of ordinary skill in the pertinent art. I understand that secondary considerations of non-obviousness, such as commercial success, long-felt but unresolved needs, failure of others, and so forth, may be assessed as well. I have been informed that an obviousness analysis must consider the full scope of the claims to avoid impermissibly using hindsight to invalidate a patent.

20. In considering whether certain prior art renders a particular patent claim obvious, I have been informed that I can consider the scope and content of the prior art, including the fact that a POSITA would regularly look to the disclosures in patents, trade publications, journal articles, conference papers, industry standards, product literature and documentation, texts describing competitive technologies,

requests for comment published by standard-setting organizations, and materials from industry conferences, as examples.

21. I have been informed that for a prior-art reference to be proper for use in an obviousness analysis, the reference must be “analogous art” to the claimed invention. A reference is analogous art if: (1) the reference is from the same field of endeavor as the claimed invention (even if it addresses a different problem); or (2) the reference is reasonably pertinent to the problem faced by the inventor (even if it is not in the same field of endeavor as the claimed invention). For a reference to be “reasonably pertinent” to the problem, it must logically have commended itself to an inventor’s attention in considering the problem. In determining whether a reference is reasonably pertinent, one should consider the problem faced by the inventor, as reflected either explicitly or implicitly, in the specification. I believe that the documents I considered in forming my opinions in this IPR are well within the range of documents a POSITA would have consulted to address the type of problems described in the Challenged Claims.

22. I have been informed that to establish that a claimed invention was obvious based on a combination of prior-art elements, an articulation of the reason(s) why a claimed invention would have been obvious must be provided. Specifically, I have been informed that the prior art, either as a single reference or a combination of multiple items of prior art, renders a patent claim obvious when there was an

apparent reason for a POSITA, at the time of the invention, to combine or modify the prior art. Rationales for combining or modifying the prior art include, but are not limited to, any of the following: (A) combining prior-art methods according to known methods to yield predictable results; (B) substituting one known element for another to obtain predictable results; (C) using a known technique to improve a similar device in the same way; (D) applying a known technique to a known device ready for improvement to yield predictable results; (E) trying a finite number of identified, predictable potential solutions, with a reasonable expectation of success; (F) identifying that known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art; or (G) identifying an explicit teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior-art reference or to combine the prior-art references to arrive at the claimed invention.

23. I have also been informed that where there is a motivation to combine, claims may be rejected as obvious provided a POSITA would have had a reasonable expectation of success regarding the proposed combination. I have also been informed that common sense may be considered. Common sense teaches that familiar items may have obvious uses beyond their primary purposes. I have been informed that if the combination was obvious to try (regardless of whether it was

actually tried) or leads to anticipated success, then it is likely the result of ordinary skill and common sense rather than non-obvious innovation.

24. I have been informed that the existence of an explicit teaching, suggestion, or motivation to combine known elements of the prior art is a sufficient, but not a necessary, condition to a finding of obviousness. In determining whether the subject matter of a patent claim is obvious, neither the particular motivation nor the avowed purpose described in the patent-in-suit controls. I have been further informed that the obviousness analysis may consider the effects of demands known to the technological community or present in the marketplace and the background knowledge possessed by a POSITA. These issues may be considered to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent.

25. I have been informed that it is improper to combine references where the references teach away from their combination. A reference may be said to teach away when a POSITA, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the patent applicant. I have also been informed that a reference does not teach away if it merely expresses a general preference for an alternative invention but does not criticize, discredit, or otherwise discourage investigation into the invention claimed.

26. I am informed that even if a case of obviousness is established, the final determination of obviousness must also consider “secondary considerations” if presented. Secondary considerations include: (a) commercial success of a product due to the merits of the claimed invention; (b) a long-felt, but unsatisfied need for the invention; (c) failure of others to find the solution provided by the claimed invention; (d) deliberate copying of the invention by others; (e) unexpected results achieved by the invention; (f) praise of the invention by others skilled in the art; (g) lack of independent simultaneous invention within a comparatively short space of time; and (h) teaching away from the invention in the prior art.

27. I have been further informed that secondary-considerations evidence is only relevant if the offering party establishes a connection, or nexus, between the evidence and the claimed invention. The nexus cannot be based on prior-art features. The establishment of a nexus is a question of fact. While I understand that Patent Owner here has not offered any secondary considerations at this time, I will supplement my opinions should Patent Owner raise secondary considerations during the course of this proceeding.

C. Claim Construction

28. I understand that the claim terms in an IPR proceeding are construed according to their plain and ordinary meaning as understood in light of the claim language, the patent’s description, and the prosecution history viewed from the

perspective of a POSITA. I further understand that where a patent defines claim language, the definition of the patent controls, even if there are other definitions that might be understood by those working in the art.

29. I understand that Petitioner contends that no express construction of any claim term is necessary in this proceeding. In forming my opinions, I have applied the ordinary and customary meaning of each claim term as a POSITA would have understood it as of January 28, 2009, in light of the '777 Patent's specification and prosecution history. Where the specification or prosecution history informs the ordinary meaning of a term, I address it in my analysis of the grounds below.

30. I have also been informed that claims 2–3, 7, 8, 9, 11, 12, 14, 18, 20, and 27–33 recite limitations in the disjunctive (for example, by listing alternatives followed by “or a combination of these”), and that such a limitation is met when the prior art teaches any one of the listed alternatives. For those limitations, my analysis addresses at least one listed alternative.

III. OVERVIEW OF THE '777 PATENT

A. Summary of the '777 Patent

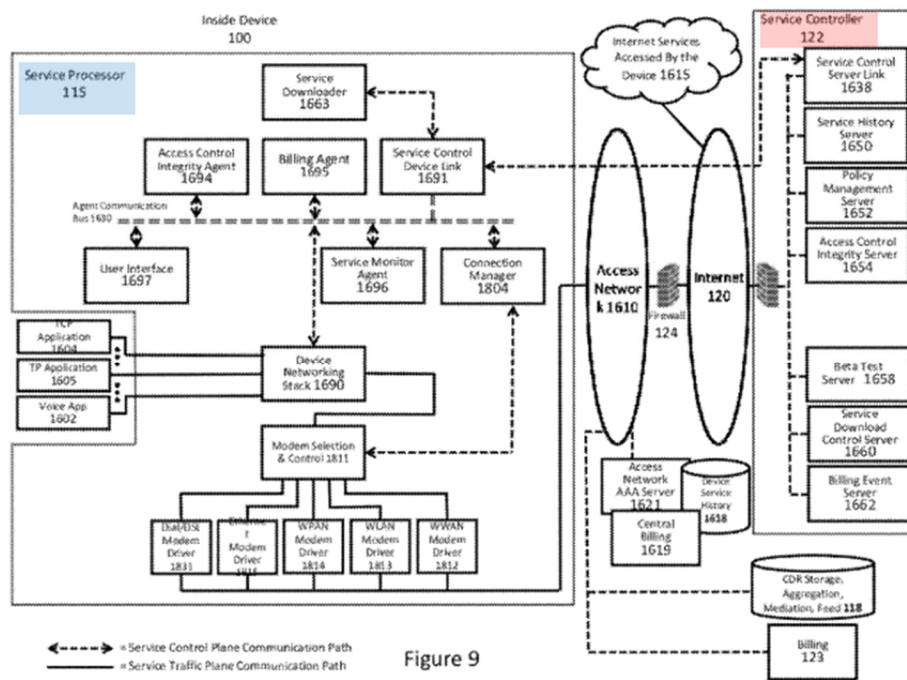
31. The '777 Patent, titled “Adapting Network Policies Based on Device Service Processor Configuration,” concerns authenticating an end-user device's service processor, which comprises device agents that assist in implementing a service policy. '777 Patent, Title, 46:57–60, 48:65–49:19, 58:3–11. The '777 Patent

explains that a device agent that can assist the network in policy implementation or enforcement “may be termed a ‘device policy implementation agent,’ which in some embodiments may be part of the service processor.” *Id.*, 46:57–60. In some embodiments, the end-user device has a service processor (e.g., a device client) “configured to implement or enforce a device-based portion of a wireless access network service policy and communicate with a network-based service controller in order to provide service processor authentication information configured to allow verification that the service processor is present and properly configured on the device.” *Id.*, 48:65–49:19. The service controller, in turn, “performs authentication of the service processor to ensure that it is present and properly configured to implement a device portion of an access network service policy.” *Id.*, 58:3–11. A POSITA would have understood from these passages that the ’777 Patent concerns authenticating the service processor—a device-side component, which can be implemented in software, made up of device agents that help implement a service policy—so that the network can verify that the service processor is present and properly configured on the device.

32. The ’777 Patent also explains that the service processor can be implemented as software that is downloaded to the device. It states that “all or a portion of the service processor 115 functions disclosed herein are implemented in software,” and that “all or substantially all of the service processor 115 functionality

(as discussed herein) is implemented and stored in software that can be performed on (e.g., executed by) various components in device 100.” ’777 Patent, 30:4–11. Service controller 122’s service download control server 1660 “provides a download function to install and/or update service software elements (e.g., the service processor 115 and/or agents/components of the service processor 115) on the device.” *Id.*, 26:45–51.

33. Figure 9, reproduced below, illustrates the ’777 Patent’s device-side “service processor” 115 and network-side “service controller” 122. ’777 Patent, 22:49–23:9, Fig. 9. Service processor 115 operates on the end-user device and includes multiple device agents, such as an access control integrity agent, a billing agent, and a service monitor agent, as well as a service control device link that communicates with service controller 122. *Id.*, 22:49–23:9, Fig. 9. The ’777 Patent treats the division of functions among the agents as a design choice: “the functional lines can be re-drawn in any way that the product designers see fit.” *Id.*, 22:49–23:9. The ’777 Patent uses “service controller” to refer to “a service processor authentication and management system,” and the service controller’s policy management server 1652 “transmits policies to the service processor 115 via the service control link 1653.” *Id.*, 25:34–37, 48:31–34.



'777Pat, Fig. 9.³

34. Claim 1, the '777 Patent's sole independent claim, recites a method in which a network system receives a device credential from an end-user device over a wireless access network, obtains a service policy applicable to the device based at least in part on that credential, obtains authentication information for the device's service processor—a service processor “comprising one or more device agents at least assisting to implement the service policy”—and uses that information to facilitate execution of an end-user device service processor authentication procedure. '777 Patent, 68:25–40. Claims 2–33 depend, directly or indirectly, from

³ All color annotations added unless otherwise noted.

claim 1. I quote claims 8 and 19 as corrected by the March 13, 2018 Certificate of Correction included in Ex. 1001, which changed “associating first” to “associating the first” in claim 8 (8[b]) and “report is from” to “report from” in claim 19.

B. Field of Endeavor

35. I have been informed that the field of endeavor of the claimed invention can be determined by reference to explanations of the invention’s subject matter in the patent application, including the embodiments, function, and structure of the claimed invention.

36. The ’777 Patent’s claims recite a network system that receives a device credential from an end-user device over a wireless access network, obtains the service policy applicable to that device, and uses service processor authentication information to facilitate authentication of the device’s service processor. ’777 Patent, 68:25–40. The specification describes an end-user device whose service processor implements or enforces a device-based portion of a wireless access network service policy; device credential sources that provide the device’s credential to network elements “for the purpose of seeking or gaining admission to the wireless access network”; and a service controller that authenticates the service processor, checks that it reports the device’s usage in the expected manner, and checks those reports against reports from a trusted source. *Id.*, 48:9–17, 48:65–49:19, 58:3–11. The policies that the service processor helps implement include access control, service

usage limit, service accounting or charging, and service notification policies. *Id.*, 48:24–31. For these reasons, in my opinion, the field of endeavor of the '777 Patent is managing an end-user device's access to and use of network services. As I explain in the next section and in my overview of each reference, Wright, Bittmann, and Durham are each in that field.

C. Problem Allegedly Solved by the Inventors of the '777 Patent

37. I have been informed that a reference is reasonably pertinent to the problem faced by the inventor if, even though it may be in a different field of endeavor, it logically would have commended itself to an inventor's attention in considering that problem, as the problem is reflected, explicitly or implicitly, in the specification. I have therefore analyzed the '777 Patent to identify the problem its inventors addressed.

38. The '777 Patent addresses a problem that arises when an end-user device carries part of the responsibility for implementing a service policy. It explains that “[i]f the device agents required to implement the device portion of the access network service policy are not present on the device or are not properly configured, then the overall combined network service policy can be in error or may not be possible to achieve.” '777 Patent, 46:46–50. It describes embodiments that “prevent, detect, or take action in response to moving a device credential from one device to another, improper configuration of a service processor, a missing service processor,

or tampering with a service processor.” *Id.*, 48:3–8. It also explains that if a device reports its data usage to the network, “a user might attempt to hack the device so that its reports contain information that is more favorable to the user than it should be, e.g., by reporting less data usage than the device actually used.” *Id.*, 47:41–45. In response, the service controller “authenticates the service processor and checks that it is reporting the end-user device’s usage in the expected manner,” and it checks “whether the reports sent by the service processor are consistent with reports from a trusted source, such as a network element.” *Id.*, 48:9–17. The service processor provides “service processor authentication information configured to allow verification that the service processor is present and properly configured on the device,” and the service controller authenticates the service processor “to ensure that it is present and properly configured to implement a device portion of an access network service policy.” *Id.*, 48:65–49:19, 58:3–11. For these reasons, in my opinion, a problem the ’777 Patent addresses is verifying that device-side components implementing a service policy are present, properly configured, and accurately reporting usage.

39. In my opinion, Wright, Bittmann, and Durham are each analogous art to the ’777 Patent. Each is in the same field of endeavor—managing an end-user device’s access to and use of network services—and each is reasonably pertinent to the problem of verifying that device-side components implementing a service policy

are present, properly configured, and accurately reporting usage. Wright describes administering security policies for mobile devices, including a security policy that “may not allow a particular network service or application or both to be used.” Wright, [0011], [0014]. Wright’s authorization module 232 authorizes a communication exchange between the client mobile device and the policy distribution or policy management modules as “a further safeguard against unauthorized or rogue mobile devices trying to hijack the security policies or corporate data,” for example by using “a cryptographic authentication protocol.” *Id.*, [0052]. Computer system 200 also authenticates client system 201 with a key provisioned to the enterprise’s managed clients: “Clients installed by some other enterprise will have a different key, and clients installed in an unmanaged mode will not have a key.” *Id.*, [0130], [0133]–[0137]. Wright thus addresses whether the network can trust the device-side software that receives and enforces its policies. Bittmann relates to “management of a communication network account”: an activity monitoring unit installed in or connected to the device monitors and reports the device’s data activity to a server utility, which rates the activity, updates the account’s allowances, and changes the permitted activity when an allowance reaches a predetermined restriction level. Bittmann, [0001], [0008]. Bittmann’s server utility can correlate the device’s reports with “other reports of activity” from “multiple sources,” and Bittmann seeks “to ensure integrity of the activity data” the device

reports, through “safety features intended to block the ability to tamper with the device or with the automatic transmission feature” and through reporting “in a manner not controlled by the user.” *Id.*, [0042], [0057], [0062]. Bittmann thus addresses whether a device-side monitor accurately reports the device’s usage. Durham relates to “enforcement of network access policies” and describes a firmware agent that operates “as a tamper-resistant agent on a host platform” so that it can “enforce policies even when the host operating system is compromised,” and its network access PDP 150 sends a nonce “to establish proof-of-liveness for firmware agent 125.” Durham, [0001], [0010], [0027]. Durham thus addresses whether the device-side component that enforces the network’s policy is present and operating as configured. These teachings would have logically commended themselves to an inventor considering the problem the ’777 Patent addresses. I explain each reference’s field and pertinence further in my overview of that reference.

D. File History of the ’777 Patent

40. The ’777 Patent issued from Application No. 13/237,827, filed September 20, 2011. ’777 Patent, cover. The earliest claimed priority date is January 28, 2009, the filing date of U.S. Provisional Application No. 61/206,354. *Id.* For purposes of this proceeding, I assume that date without offering an opinion that each claim is entitled to it.

41. Original claim 1 recited a “network system” comprising “one or more network elements configured to” receive a device credential from an end-user device and “a service processor authentication and management system” configured to obtain service processor authentication information and, using that information, facilitate execution of an end-user device service processor authentication procedure. Ex. 1003, 133.

42. After Applicant responded to an indefiniteness rejection, the Examiner, on December 31, 2013, withdrew that rejection and rejected the pending system claims under pre-AIA 35 U.S.C. § 103(a) over U.S. Patent No. 8,185,127 (“Cai”). Ex. 1003, 2109–13. The Examiner treated the “enabling” and “configured to” language of claim 1, as then amended, as intended-use language that was “not given patentable weight,” read the claims broadly on that basis, and found it obvious for Cai to include a communication interface and network elements configured to perform the recited functions. *Id.* On March 28, 2014, Applicant converted the claims from system claims to method claims. *Id.*, 2210–16. Applicant then distinguished Cai on the ground that Cai did not disclose or suggest a “service processor” on an end-user device and therefore did not disclose or suggest either “obtaining service processor authentication information associated with a service processor on the end-user device” or “using the service processor authentication

information, facilitating execution of an end-user device service processor authentication procedure.” *Id.*, 2217–22.

43. On May 9, 2014, following a telephone interview with Applicant’s counsel, the Examiner entered an Examiner’s amendment that added to the step of obtaining service processor authentication information the limitation “the service processor comprising one or more device agents at least assisting to implement the service policy,” now 1[c][ii]. Ex. 1003, 2231–33. The Notice of Allowance states that the claims are allowed pursuant to Applicant’s March 28, 2014 amendment and remarks, and it gives as the Examiner’s statement of reasons for allowance that “[t]he prior art of record, either alone or in combination, does not disclose, or suggest, Applicant’s claim language.” *Id.*, 2232.

44. The ’777 Patent lists hundreds of references cited during its prosecution. ’777 Patent, (56). Among them are references from Wright’s patent family. On April 12, 2012, Applicant submitted an information disclosure statement listing U.S. Patent No. 7,353,533 to Wright et al. (“the ’533 patent”) and U.S. Patent Application Publication No. 2008/0109679 A1 to Wright et al. (“the ’679 publication”). Ex. 1003, 222, 230. The ’533 patent issued from Application No. 10/413,443, the application that published as Wright, and the ’679 publication identifies itself as a divisional of that application. Ex. 1007, cover; Ex. 1008, [0001]; Wright, cover. I have compared the ’533 patent with Wright. The ’533 patent

contains the same written description as Wright, including the descriptions of client system 201 and its modules, the encrypted token, the enterprise key, and the “Establish Session” protocol on which I rely in this declaration. On September 9, 2013, Applicant submitted another information disclosure statement identifying U.S. Patent No. 7,478,420 to Wright et al. (“the ’420 patent”), whose related-application data lists a continuation-in-part of Application No. 10/413,443. Ex. 1003, 1798; Ex. 1009, cover. The Examiner marked each of these references as considered, without discussing them. Ex. 1003, 1554, 1562, 2115.

45. In my opinion, Wright, whose written description the ’533 patent shares, teaches the limitations on which Applicant relied to overcome Cai and the limitation that the Examiner’s amendment added. Applicant argued that Cai did not disclose or suggest a service processor on the end-user device, *obtaining service processor authentication information associated with a service processor on the end-user device* (1[c][i]), or *using the service processor authentication information, facilitating execution of an end-user device service processor authentication procedure* (1[d]); the Examiner’s amendment then added *the service processor comprising one or more device agents at least assisting to implement the service policy* (1[c][ii]). Wright discloses or suggests each of these features. Client system 201 is software that executes on the mobile device and “enforces the policies set up by the administrator” (a *service processor on the end-user device*). Wright, [0046].

Its policy setting module 212, policy enforcement control module 214, and authorization module 245 receive, install, and enforce the administrator's policies and send authorization information to computer system 200 (*one or more device agents at least assisting to implement the service policy*). *Id.*, [0058]–[0059], [0061], [0064], [0068]. In Wright's "Establish Session" protocol, client system 201 computes "SHA1(random challenge, key)" on the mobile device and returns it to computer system 200, which receives it (*obtaining service processor authentication information associated with a service processor on the end-user device*), and computer system 200 "computes the same SHA1 hash" and compares it with the client's response to determine whether the client is authenticated (*using the service processor authentication information, facilitating execution of an end-user device service processor authentication procedure*). *Id.*, [0133]–[0137]. I explain these teachings in detail in my analysis of 1[c][i], 1[c][ii], and 1[d] in Ground 1A. For the reasons explained in this declaration, Wright alone renders obvious claim 1 and claims 2–7, 10, 11, 13, and 14, and Wright renders obvious the remaining Challenged Claims in view of Bittmann, Durham, or both. Neither Bittmann nor Durham is among the references cited on the face of the '777 Patent. '777 Patent, (56).

IV. LEVEL OF A PERSON OF ORDINARY SKILL

46. Based on my review and analysis of the '777 Patent, the prior art cited herein, and the ordinary-skill factors described above, it is my opinion that a POSITA relating to the subject matter of the '777 Patent as of January 28, 2009 would have had (1) at least a bachelor's degree in computer science, electrical engineering, or a related field, and (2) 3-5 years of experience in services and application implementation in communication networks. Additional graduate education could substitute for professional experience, and vice versa. This level of skill reflects the subject matter of the '777 Patent and the prior art, which concern software on end-user devices and servers in communication networks that together implement, enforce, and account for network services and the policies that govern them. When I refer to the understanding of a POSITA, I am referring to the understanding of such a person as of January 28, 2009.

47. As of January 28, 2009, I had more than ordinary skill in the art. I am, however, familiar with the skills and knowledge possessed by those I would have considered to be of ordinary skill in the art as of that date. In the late 1990s, for example, my team and I at China Lake implemented a demonstration version of our wavelet-based video communications system that used network communications (TCP/IP and UDP over the Internet) to transfer the compressed video in real time from the encoder to the decoder and to send camera-control signals back the other

way. Later, after arriving at NMSU, I led an ARO-funded program that developed a distributed coding testbed that used Bluetooth and WiFi radios to share data between nodes and to coordinate their efforts for maximum bandwidth efficiency. I advised a number of faculty on various projects related to the use of network IP protocols to facilitate the real-time transmission of data between different programs running on multiple networked computers. For instance, I worked closely with colleagues at NMSU whose research specialties focused on secure network communications prior to 2009, including my former colleague, the late Dr. Hong Huang. In that same timeframe, I also taught a graduate course in Telemetry Systems that introduced students to the iNet military standard for wirelessly networking devices (aircraft and test articles), which supported the use of TCP/IP protocols over various military radios. The China Lake video communications system and the NMSU distributed coding testbed each involved implementing applications that operate over communication networks, which is the kind of experience described in the level of ordinary skill above.

48. My opinions provided in this declaration would not change in view of minor modifications to this level of ordinary skill.

V. OVERVIEW OF THE TECHNOLOGY

49. I was asked to summarize the technology relevant to the '777 Patent. This section summarizes that technology as the '777 Patent itself describes it. I

describe Wright, Bittmann, and Durham in the next section, and I apply them to the Challenged Claims in the Grounds section of this declaration.

A. Device Agents

50. As I explained in the Summary of the '777 Patent section above, the '777 Patent describes software on an end-user device that helps carry out a service policy: service processor 115, which comprises device agents such as an access control integrity agent, a billing agent, and a service monitor agent, and whose functionality can be “implemented and stored in software.” '777 Patent, 22:49–23:9, 30:4–11, Fig. 9.

B. Device Credentials and Network Admission

51. The '777 Patent describes networks that identify a device seeking service by a credential associated with the device and that use network elements to decide whether to admit it. Its examples of such credentials include information from a SIM card, an IMSI, an IMEI or MEID, a MAC or IP address, a secure device identifier, or a device encryption key, which the device provides “to one or more network service policy enforcement elements (e.g., AAA, HLR, PCRF, access network authentication system, admission system or log-in system) for the purpose of seeking or gaining admission to the wireless access network.” '777 Patent, 49:2–12. It refers to AAA and HLR elements as part of the existing network, describing new service usage measures and billing capabilities that are “desirable to implement

in a manner that does not require major changes or upgrades to the existing network gateways, routers, MWC/HLRs, AAA, CDR aggregation, CDR mediation, billing and/or provisioning systems.” *Id.*, 10:44–49.

C. Service Processor Authentication

52. The ’777 Patent distinguishes the device credential from the information used to authenticate the service processor. The device provides its device credential to network elements “for the purpose of seeking or gaining admission to the wireless access network,” while the service processor communicates with the service controller “in order to provide service processor authentication information configured to allow verification that the service processor is present and properly configured on the device,” and the service controller uses that information to verify the service processor. ’777 Patent, 49:2–12, 49:15–19, 49:31–35; see also *id.*, 48:9–13 (the service controller “authenticates the service processor”).

D. Network Records of Device Usage

53. The ’777 Patent also describes records of a device’s usage that network functions generate. It explains that a charging data record (CDR) is “typically generated by one or more network functions that supervise, monitor, and/or control network access for the device,” and that CDRs “typically form the basis for recording device network service usage, and often form the basis for billing for such

usage.” ’777 Patent, 9:56–62. It also describes checking a device’s reports against such records: its service controller determines whether the service processor’s reports are “consistent with reports from a trusted source, such as a network element,” and a trusted source can be “a network-based element such as a server, gateway, proxy or router.” *Id.*, 48:14–17, 64:37–45.

VI. SUMMARY OF THE PRIOR ART REFERENCES

A. Overview of Wright

54. U.S. Patent Application Publication No. 2004/0123153 A1 to Wright et al. (“Wright”) was filed April 11, 2003 and published June 24, 2004, before the assumed January 28, 2009 priority date of the ’777 Patent. Wright, cover.

55. Wright describes an enterprise client-server system that administers security policies for mobile devices, including policies governing which network services and applications a device may use. Wright, Abstract, [0011], [0014]. Wright summarizes its subject as “[t]he administration of protection of data on a client mobile computing device by a server computer system such as within an enterprise network or on a separate mobile computing device.” *Id.*, Abstract. A policy may govern the device’s use of network services and applications: “a security policy may not allow a particular network service or application or both to be used based upon either or both of a particular detected location or the activity status of a security feature.” *Id.*, [0014]. Policies can apply to groups of users or to “the mobile devices

themselves.” *Id.*, [0049]. The mobile devices include notebook computers and personal digital assistants that connect to networks through wired connections or through wireless access points. *Id.*, [0013], [0044]–[0045], Fig. 1. Wright’s server-side computer system 200 (“system 200”) manages and distributes the policies, and client system 201 (“client 201”), software executing on the mobile device, enforces them. *Id.*, [0046].

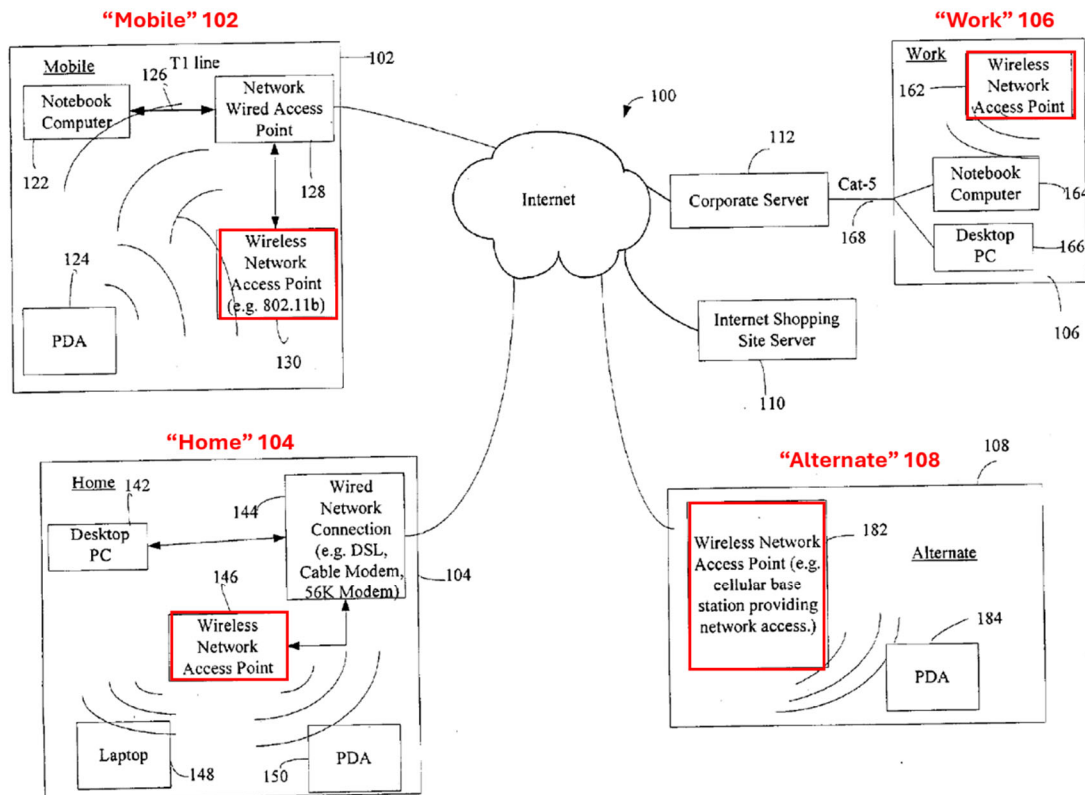
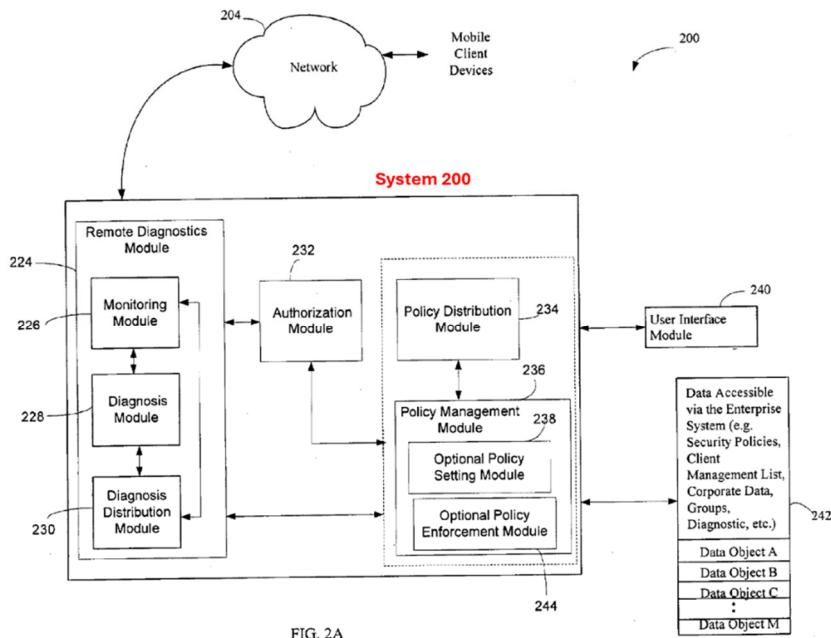


FIG. 1

Wright, Fig. 1.

56. Figure 1 illustrates the network environments in which Wright’s mobile devices operate, at locations designated “Home,” “Work,” “Mobile,” and

“Alternate,” each connect to the Internet. Wright, [0042]–[0045], Fig. 1. The mobile devices reach those networks through wired connections or through wireless network access points (WAPs). At the “Home” location, laptop 148 and PDA 150 communicate through wireless network access point 146; at the “Work” location, notebook computer 164 can communicate through wireless NAP 162; at an airport (“Mobile”), notebook computer 122 and personal digital assistant 124 “have a wireless connection to a wireless NAP 130, in this example an 802.11b connection”; and at the “Alternate” location, wireless network access point 182 “is associated with a cellular base station providing network access through a General Packet Radio Services (GPRS) system, Global System for Mobile communication (GSM) system, third generation wireless 3G system or other kind of mobile wireless communication system.” *Id.*, [0042]–[0045], Fig. 1. The security policy for a location can also take into account whether the device’s connection is wired or wireless. *Id.*, [0044].



Wright, Fig. 2A.

57. Figure 2A depicts system 200, which “allows an administrator to manage and distribute policies and software upgrades, analyze logs, and perform remote diagnostics.” Wright, [0046]. System 200 may execute on a standalone computer or “be implemented by a series of networked computers as may typically be implemented by an enterprise.” *Id.* It comprises authorization module 232, policy distribution module 234, policy management module 236, remote diagnostics module 224, and user interface module 240, each with access to data objects stored in memory 242 and to “a network 204 (e.g. Internet).” *Id.*, [0047]. Policy management module 236 manages security policies, including defining each policy’s “rules and permissions,” the ports, network services, and applications to be monitored, and the “enforcement mechanisms to be put in place for a particular

policy.” *Id.*, [0048]. Policy distribution module 234 “distributes security information to the one or more client mobile devices,” and authorization module 232 “authorizes a communication exchange between the client mobile device and the policy distribution or policy management modules.” *Id.*, [0051]–[0052].

58. Policy management module 236 also associates each policy with the devices to which it applies, provisions those devices, and keeps a record for each device. Module 236 “designates 304 one or more client devices associated with the policy,” and the association may be based on a group whose member is a user or the device itself: “Another example of a member is the mobile device itself.” Wright, [0078]. “In a secure manner,” module 236 then provides the designated devices with “authorization information for use in contacting the server system securely” (step 306). *Id.* Wright’s example of that authorization information is “an encrypted token provided by the authorization module 232 to the mobile client device during a trusted connection between the two,” such as “an internal connection behind the firewall of and within the internal network of the enterprise.” *Id.* Where security information is to be encrypted, module 236 also provides the device with “cryptographic information which the client device can store and use to decrypt the security information,” for example “a key for use with a cryptographic authentication protocol” (step 308). *Id.* Module 236 “maintains 314 client management information for the mobile device and the one or more policies associated with it,” organized in

a data object accessible to system 200. *Id.*, [0104], [0120]. Through system 200's user interface, the administrator manages that information client device by client device: Wright's management interface lists "all mobile client devices that have connected to the server system," and its actions include "Reassign this client device to another group." *Id.*, [0121], [0126].

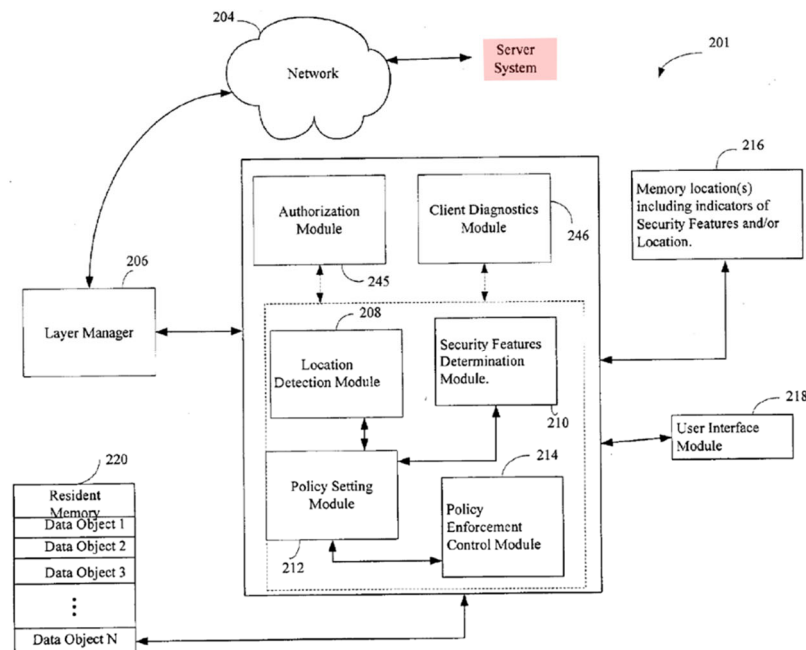
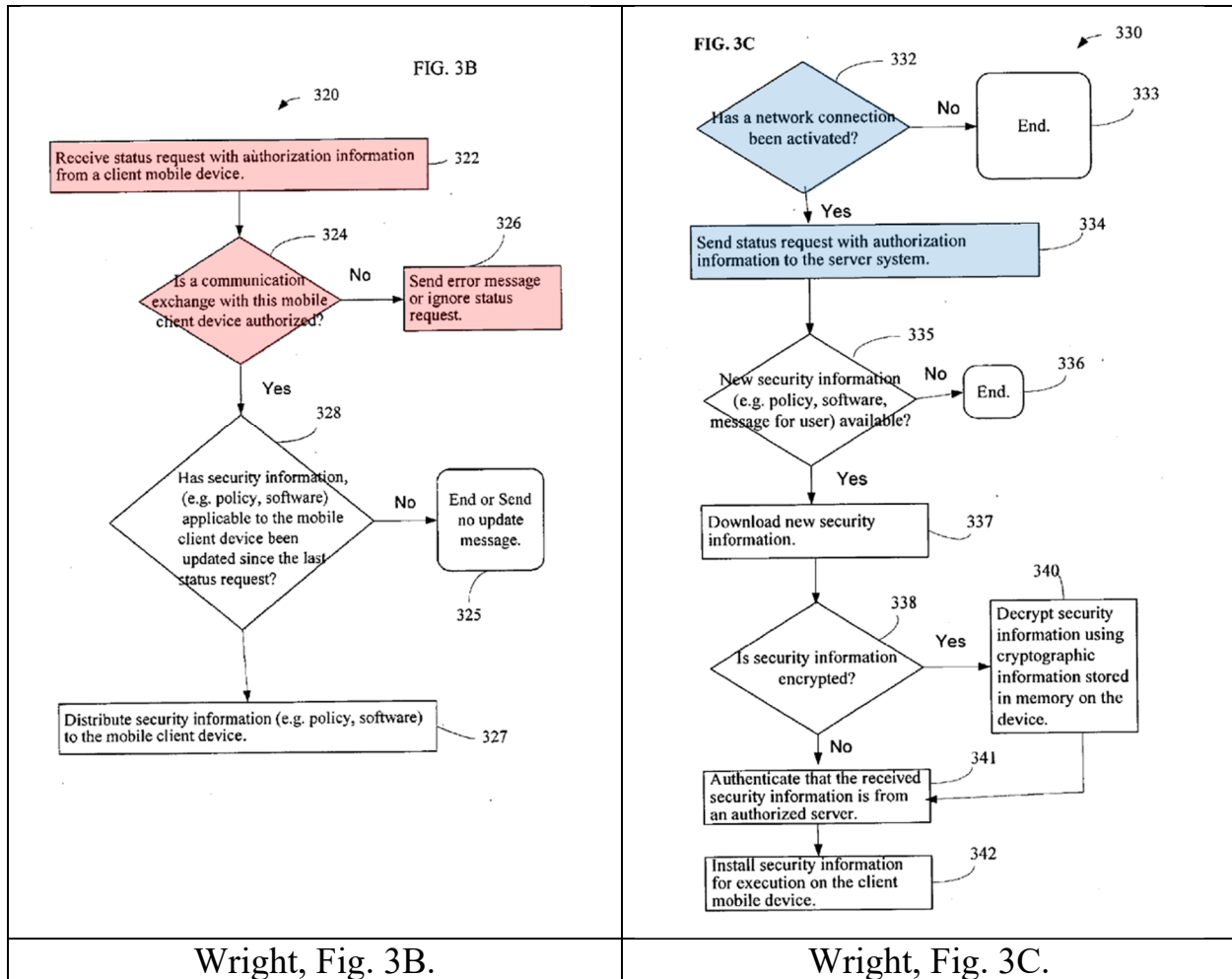


FIG. 2B

Wright, Fig. 2B.

59. Figure 2B depicts client 201, which “may be implemented as software executing in a mobile computing device” and “enforces the policies set up by the administrator.” Wright, [0046]. Client 201 comprises location detection module 208, policy setting module 212, security features determination module 210, policy enforcement control module 214, layer manager 206, user interface module 218,

authorization module 245, and client diagnostics module 246. *Id.*, [0058]. Authorization module 245 “provides authorization information to the authorization module 232 of the server system 200 to establish communication exchanges with the client mobile device.” *Id.*, [0059]. Policy setting module 212 “receives, installs and updates” the security policies that system 200 distributes over network 204, and it “communicates the current security policy to the policy enforcement control module 214.” *Id.*, [0061], [0068]. Module 214 “comprises instructions for enforcing the security policy currently set by the policy setting module 212” and “comprises one or more enforcement mechanism modules as specified by the policy”; for example, it may prevent certain files from being transferred over a wireless connection. *Id.*, [0064], [0068]. Client diagnostics module 246 collects diagnostic information that is sent to remote diagnostics module 224 of system 200. *Id.*, [0059].



60. Client 201 begins its exchanges with system 200 by sending a status request. Each client mobile device “sends a status request upon network access starting up to the server system,” and “[t]he status request provides information regarding the client mobile device, examples of which are authorization information, policies and their version numbers resident on the device, and diagnostic information.” Wright, [0128]. As Figure 3B shows, authorization module 232 “receives 322 the status request with authorization information from a client mobile

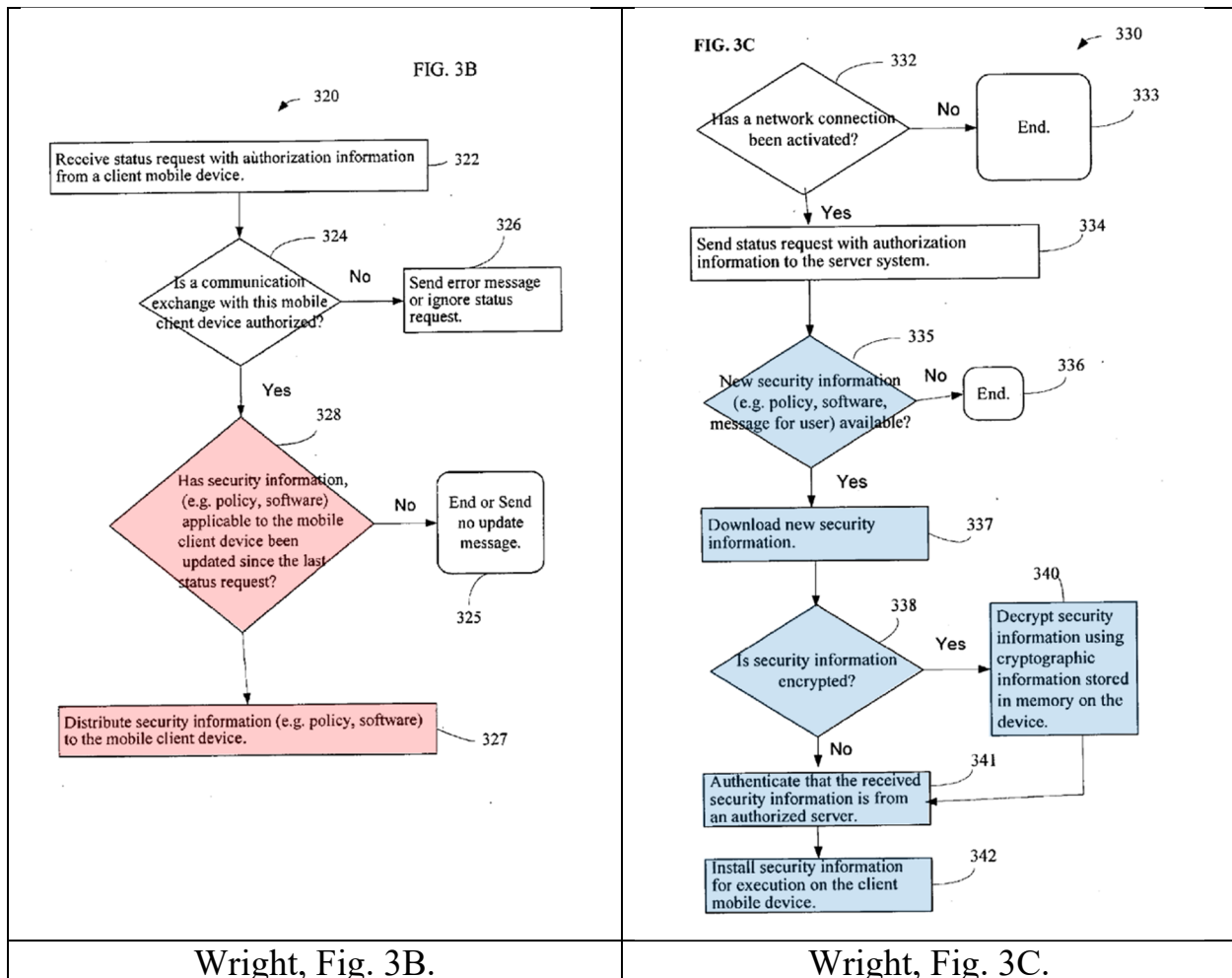
device, and it determines 324 whether a communication exchange with this mobile device is authorized.” *Id.* If the exchange is not authorized, “the status request is ignored 326 or an error message is sent 326.” *Id.* If it is authorized, policy distribution module 234 “determines 328 if security information (e.g. a policy or software) applicable to the mobile client device has been updated since the last status request”; if so, module 234 “distributes 327 the updated security information to the mobile device over the network (e.g. Internet) 204,” and if not, “the status request is ignored 325 or a no update message is sent 325.” *Id.*, Fig. 3B. Wright’s example of the authorization information that system 200 provisions to a device “for use in contacting the server system securely” is the encrypted token, and, as I explain for 1[a] in Ground 1A, a POSITA would have understood that the authorization information accompanying the status request includes that token. *Id.*, [0078], [0128]–[0129].

61. For the authorization decision, Wright states that “[v]arious authorization protocols and techniques may be used,” including “a simple username and password verification scheme” and “a cryptographic authentication protocol.” Wright, [0052]. Wright also describes the cryptographic material for such a protocol. When system 200 is installed, “a key is generated and stored in an encrypted file,” and “[t]his encrypted file is copied and distributed to the clients and any other servers in the server system.” *Id.*, [0130]. The key serves two purposes: “authentication of

clients and authentication of a downloaded security information such as policies by a client.” *Id.* “Clients installed by some other enterprise will have a different key, and clients installed in an unmanaged mode will not have a key.” *Id.* A POSITA would have understood from this description that the key is a secret common to the enterprise’s managed clients: one key is generated when system 200 is installed, the same encrypted file is copied to each client, and, in describing the key used for “authentication of clients,” Wright distinguishes keys by enterprise and by managed or unmanaged installation, not by individual client. *Id.*

62. Wright describes one such cryptographic authentication protocol, the “Establish Session” challenge-response protocol, which uses that key. Wright, [0130], [0133]–[0138]. Wright introduces it as an “alternate embodiment” in which “a session may be established between the server system 200 and the client system 201 using protocols,” and as a protocol “that may be used to authenticate the client as a first step in all client-server transactions.” *Id.*, [0133]. In the first message, client 201 sends system 200 its “protocol version” and user identification (“UID”); in the second, system 200 sends a “random challenge”; and in the third, client 201 returns “SHA1(random challenge, key).” *Id.*, [0134]–[0136]. System 200 “computes the same SHA1 hash,” and “if it doesn’t match what the client sent, then the server drops the connection with no further ado”; “[o]therwise, after this protocol is finished, the server has authenticated the client.” *Id.*, [0137]. If a session is not authorized or

established, client diagnostics module 246 “creates 328 an audit event noting failure to establish the session”; if a session is established, policy setting module 212 sends an update query to system 200. *Id.*, [0138]. In my opinion, Establish Session is a cryptographic authentication protocol of the kind Wright identifies for authorization module 232. SHA1 is a cryptographic hash function, and the protocol proves that client 201 holds the enterprise key without sending the key over the network: only a party that holds the key can compute the value that matches system 200’s own hash of the fresh challenge. *Id.*, [0052], [0130], [0133]–[0137].



63. For an authorized and authenticated client, system 200 distributes the device's security rules when applicable, and client 201 authenticates and installs them. Wright, [0128], [0132], [0139]–[0141], [0144], [0146]–[0147]. Figure 3C shows the client side. When policy setting module 212 determines that a network connection has been activated (step 332), authorization module 245 sends a status request “checking if new security information is available along with authorization information to the server system 200” (step 334). *Id.*, [0129]. On the server side, policy distribution module 234 determines whether security information “applicable to the mobile client device” has been updated and, if so, distributes it (steps 328 and 327). *Id.*, [0128]. Wright's protocols carry out the same exchange after Establish Session: policy setting module 212 sends the “Query Policy Update” message with the client's policy version, system 200 answers either “there is a newer version” or “your version is up-to-date,” and, if a newer policy exists, client 201 sends the “Download Policy” verb and system 200 returns “the encrypted policy file.” *Id.*, [0138]–[0141], [0144]–[0147]. Client 201 decrypts a downloaded policy with its previously provisioned cryptographic information (step 340), and authorization module 245 authenticates it (step 341) by comparing a SHA1 hash of the decrypted policy with the hash that system 200 appended, so that client 201 can “safely conclude that the policy was generated on an authorized server, and that the policy

has not been corrupted (nor tampered with).” *Id.*, [0131]–[0132]. The policy is then “installed 342 for execution on the client mobile device.” *Id.*, [0132].

64. Wright also describes how system 200 learns about the condition of client 201. Client diagnostics module 246 sends diagnostic information to remote diagnostics module 224 of system 200, whose monitoring module 226 “receives diagnostic information such as events or audit logs from a client device and stores the information in a data object (242) for the client device.” Wright, [0054], [0059]. When a session is not authorized or not established, client diagnostics module 246 “creates 328 an audit event noting failure to establish the session.” *Id.*, [0138]. In another embodiment, the protective components on the device check one another’s health: the policy engine, layered service provider, file filter, and packet filter engine each report to diagnostics module 844 “whether any of the other services have been disabled or removed from the system,” and that information “may be relayed to the policy server 850 as it indicates a possible compromise of the protective system.” *Id.*, [0255].

65. As I explained in the section on the ’777 Patent’s field of endeavor, the field of endeavor of the ’777 Patent is managing an end-user device’s access to and use of network services. Wright is in that field. Wright provides for the “administration of protection of data accessible by one or more mobile devices” through security policies that system 200 defines and distributes and that client 201

installs and enforces on the device. Wright, [0011], [0014], [0046]–[0048], [0061], [0068]. Those policies govern the device’s use of network services and applications: a policy “may not allow a particular network service or application or both to be used.” *Id.*, [0014]. System 200 also controls whether a device may exchange security information with it at all: authorization module 232 “authorizes a communication exchange between the client mobile device and the policy distribution or policy management modules” based on the authorization information the device presents. *Id.*, [0052], [0078], [0128]. Accordingly, in my opinion, Wright is in the same field of endeavor as the ’777 Patent.

66. Wright is also reasonably pertinent to the problem the ’777 Patent addresses: verifying that device-side components implementing a service policy are present, properly configured, and accurately reporting usage. Wright addresses the presence-and-configuration part of that problem for client 201, the software that implements the administrator’s policies on the mobile device. Authorization module 232 is “a further safeguard against unauthorized or rogue mobile devices trying to hijack the security policies or corporate data.” Wright, [0052]. In Establish Session, system 200 verifies that the responding client holds the key provisioned to the enterprise’s managed clients, which “[c]lients installed by some other enterprise” and clients “installed in an unmanaged mode” do not have. *Id.*, [0130], [0133]–[0137]. Client 201, in turn, authenticates each downloaded policy before installing

it, so that it installs only a policy “generated on an authorized server.” *Id.*, [0131]–[0132]. And Wright’s clients report on their own condition: monitoring module 226 receives events and audit logs from the client, and Wright’s health checks report when protective services “have been disabled or removed from the system” because that condition “indicates a possible compromise of the protective system.” *Id.*, [0054], [0255]. In my opinion, these teachings would have logically commended themselves to a POSITA considering that problem.

B. Overview of Bittmann

67. U.S. Patent Application Publication No. 2004/0148237 A1 to Bittmann et al. (“Bittmann”) was filed January 29, 2003 and published July 29, 2004, before the assumed January 28, 2009 priority date of the ’777 Patent. Bittmann, cover.

68. Bittmann describes a client-server system that monitors a communication device’s network activity, reports that activity to a server, and uses the reports to account for the device’s usage and to restrict further activity when an allowance of the device’s account reaches a pre-determined restriction level. Bittmann, Abstract, [0042]–[0045]. As shown in Figure 1, activity monitoring unit 130, which “may be installed within communication device 110 or may be connected thereto,” “monitors data activity of communication device 110” and reports that activity to activity monitoring server utility 150 over communication network 120. *Id.*, [0028], [0054], [0062]. Communication device 110 can be, among other things,

a cellular telephone, a computer, or a PDA, and network 120 can be of any type, “including inter-alia cellular, PSTN, Internet, Intranet.” *Id.*, [0032]–[0033]. Server utility 150 “manages the account of communication device 110” and processes the activity reports to account for the device’s usage. *Id.*, [0029], [0042]–[0043].

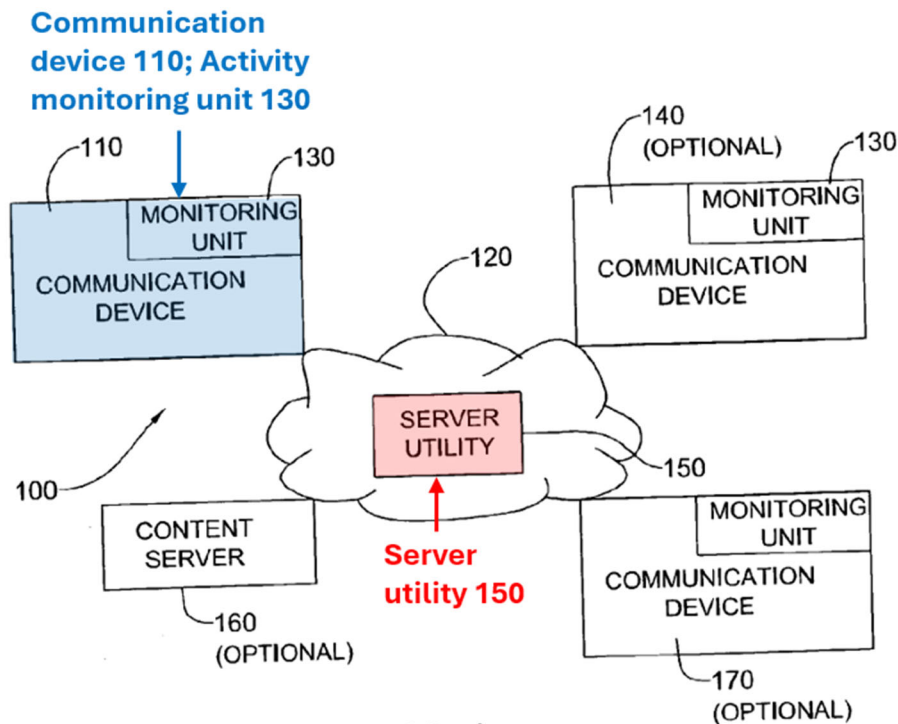


FIG. 1

Bittmann, Fig. 1.

69. Figure 4 shows monitoring unit 130 on the device side. Monitor 410 monitors the device’s data activity, and report generator 420 generates reports on that activity for server utility 150. Bittmann, [0054]. Monitor 410 “may be configured to categorize a specific data activity,” for example by “the application on communication device 110/140 using the data” or by “the source or destination of

the data transferred to or from device 110/140,” and it passes information on the categorized activity to report generator 420. *Id.*, [0059]. Monitoring unit 130 sits between the device’s CPU and its network interface, or on the bus that connects them, so that the data the device sends and receives passes where monitor 410 can observe it. *Id.*, [0052]–[0053]. Optional control 430 “can block some or all incoming data from reaching user interface 340 and/or some or all outgoing data from reaching network interface 310” when server utility 150 directs a restriction. *Id.*, [0055], [0070]–[0071].

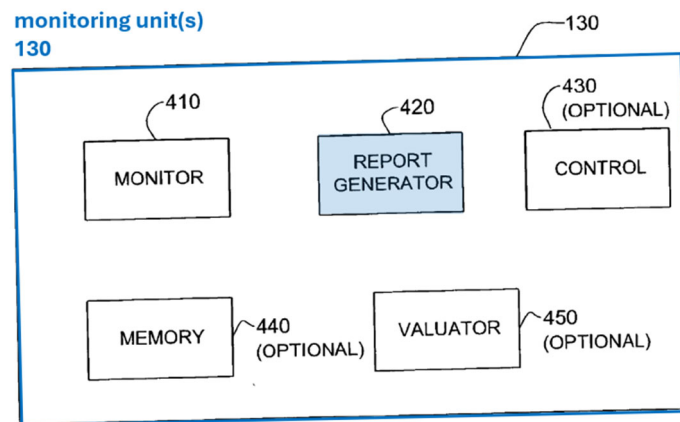


FIG. 4
Bittmann, Fig. 4.

70. Report generator 420 sends its reports to server utility 150 “automatically in a manner not controlled by the user of communication device 110,” which “ensures the integrity of activity data transfer to the server utility.”

Bittmann, [0054], [0062]. The device or monitoring unit 130 can also include “safety features intended to block the ability to tamper with the device or with the automatic transmission feature, so as to ensure integrity of the activity data transmitted to server utility 150.” *Id.*, [0057].

71. Those reports give server utility 150 the information it needs to rate the device’s activity, and Bittmann explains that reports are useful “for rating purposes and/or for fraud and network management.” Bittmann, [0059]. A report can identify, among other things, the “volume of data transferred (e.g. kilobytes, number of packets, etc),” the “time-interval for data transfer,” the “source and destination of data,” the “types of communication infrastructure utilized,” “the application on communication device 110/140 using the data,” “the application protocol,” and “any web sites accessed.” *Id.*, [0060].

72. Figure 2 shows server utility 150, which processes those reports. Mediation system 210 receives “data activity reports from monitoring units (and optionally for receiving other reports of activity),” and, if required, correlates and/or aggregates “reports from multiple sources to produce a unified report record” formatted to a protocol known to rating system 220. Bittmann, [0042]. Rating system 220 rates the reported activity using the rates, discounts, and subscription-plan information stored in database 240, and allowance manager 230 “updates the allowance(s) for the account associated with communication device 110 ... based on

the rating of activity involving communication device 110.” *Id.*, [0042]–[0043]. The rated activity includes the activity that monitoring unit 130 reports and can also include “data activity reported by system resources (for example probes, switches, VoIP servers, etc).” *Id.*, [0043]. An allowance can represent an amount of money, such as the account’s unused credit line or prepaid balance, or “a quantity of past or future activity, for example volume in bytes, volume in packets, time units, number of accesses, number of downloads”; an allowance can also represent, for example, “the number of transmitted emails.” *Id.*, [0038], [0045]. When an allowance reaches its “pre-determined restriction level,” allowance manager 230 “causes the restriction of activity in the account.” *Id.*, [0045].

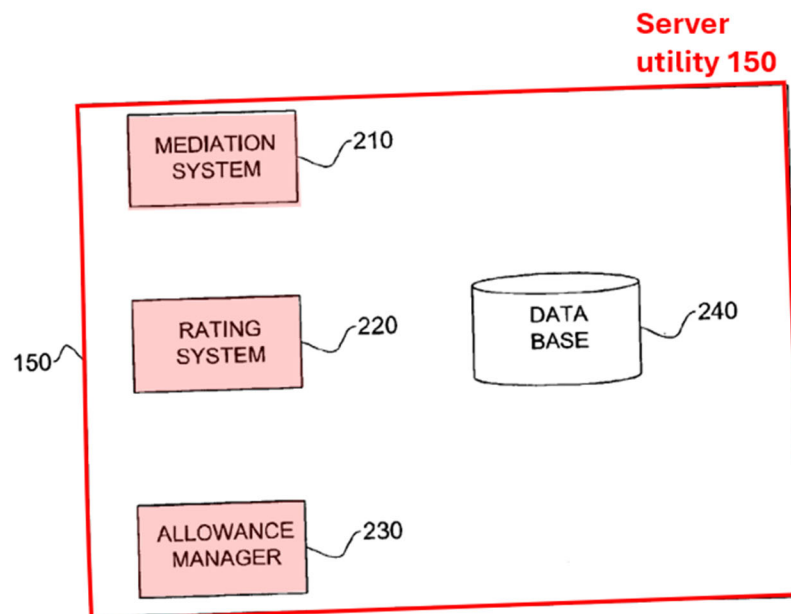
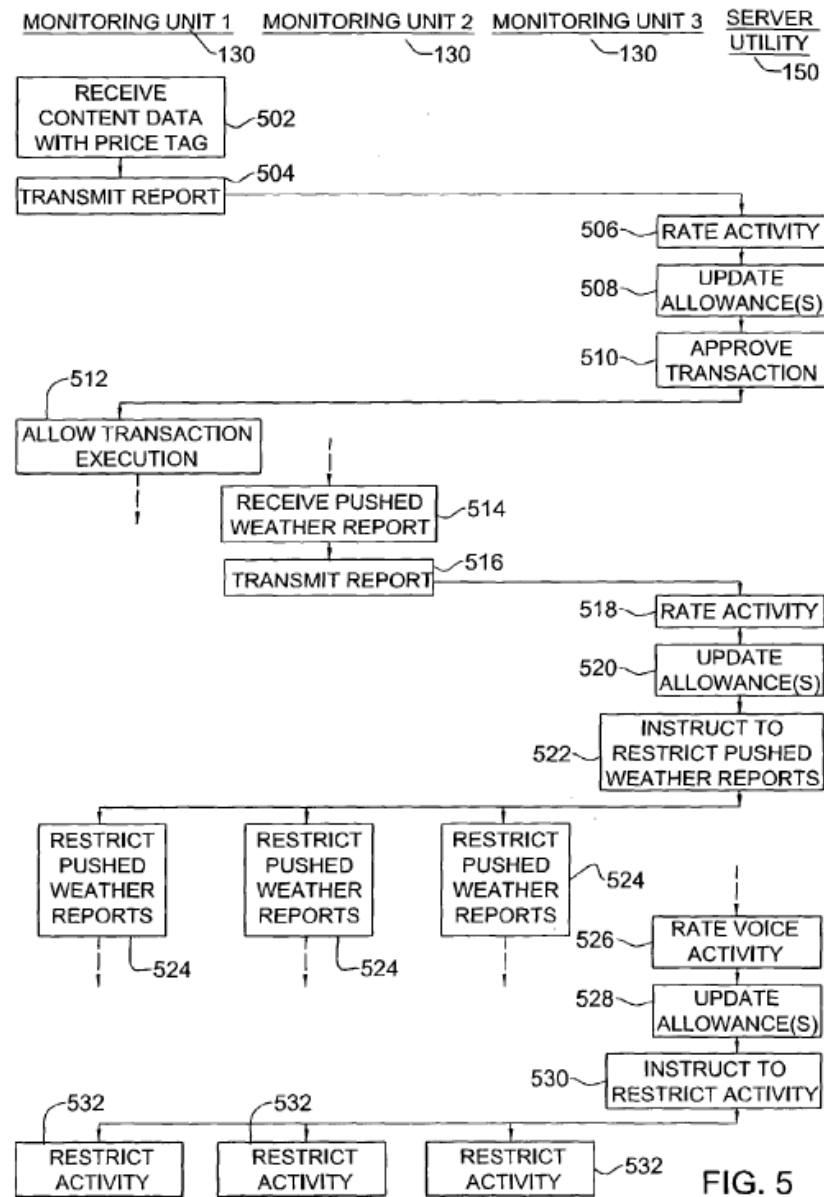


FIG. 2

Bittmann, Fig. 2.

73. Figure 5 shows these components in operation for an account with two allowances: one representing the account's unused credit line and one representing the number of pushed weather reports remaining for the day. Bittmann, [0066]. As the monitoring units on the account's devices report activity, mediation system 210 receives the reports, rating system 220 rates the reported activity, and allowance manager 230 updates the affected allowance; for example, after a reported purchase is rated at \$30, "the overall allowance for the unused credit line is reduced by \$30," and after a pushed weather report is rated, the weather-report allowance is reduced to zero. *Id.*, [0067]–[0070].

74. Figure 5 also shows server utility 150 enforcing a limit. When an allowance reaches its restriction level, allowance manager 230 "causes the restriction of activity in the account." Bittmann, [0045]. In the example, once the account's allowance of pushed weather reports is used up, server utility 150 informs the monitoring units to block further weather reports (steps 522–524); and once the credit-line allowance is overdrawn, the monitoring units "are told by server utility 150 to restrict activity, for example to only allow free traffic (e.g. toll free web sites) and traffic to the network operator," which leaves the user able to replenish the account (steps 530–532). *Id.*, [0070]–[0071]. Control 430 in each monitoring unit applies the restriction. *Id.*, [0055], [0070]–[0071].



Bittmann, Fig. 5.

75. As explained in the Field of Endeavor section, the field of endeavor of the '777 Patent is managing an end-user device's access to and use of network services. Bittmann concerns the same subject. It "relates to management of a

communication network account,” and it monitors a communication device’s data activity, rates that activity, updates the allowances of the device’s account, and changes the device’s permitted activity when an allowance reaches a pre-determined restriction level. Bittmann, [0001], [0008], [0042]–[0045]. As Figure 5 shows, server utility 150 restricts the device’s activity when an allowance is exhausted. *Id.*, [0045], [0070]–[0071]. Accordingly, in my opinion, Bittmann is in the same field of endeavor as the ’777 Patent.

76. Bittmann is also reasonably pertinent to the problem the ’777 Patent addresses: verifying that device-side components implementing a service policy are present, properly configured, and accurately reporting usage. ’777 Patent, 48:9–17, 48:65–49:19, 58:3–11. The ’777 Patent’s service controller, for example, “checks whether the reports sent by the service processor are consistent with reports from a trusted source, such as a network element.” *Id.*, 48:14–17. Bittmann confronts the same concern for the device-side component that reports usage. Monitoring unit 130 on the device monitors data activity and reports it to server utility 150, which rates the activity and manages the account. Bittmann, [0008], [0042]. Report generator 420 sends its reports “automatically in a manner not controlled by the user of communication device 110,” which “ensures the integrity of activity data transfer to the server utility,” and the device or monitoring unit 130 can include “safety features intended to block the ability to tamper with the device or with the automatic

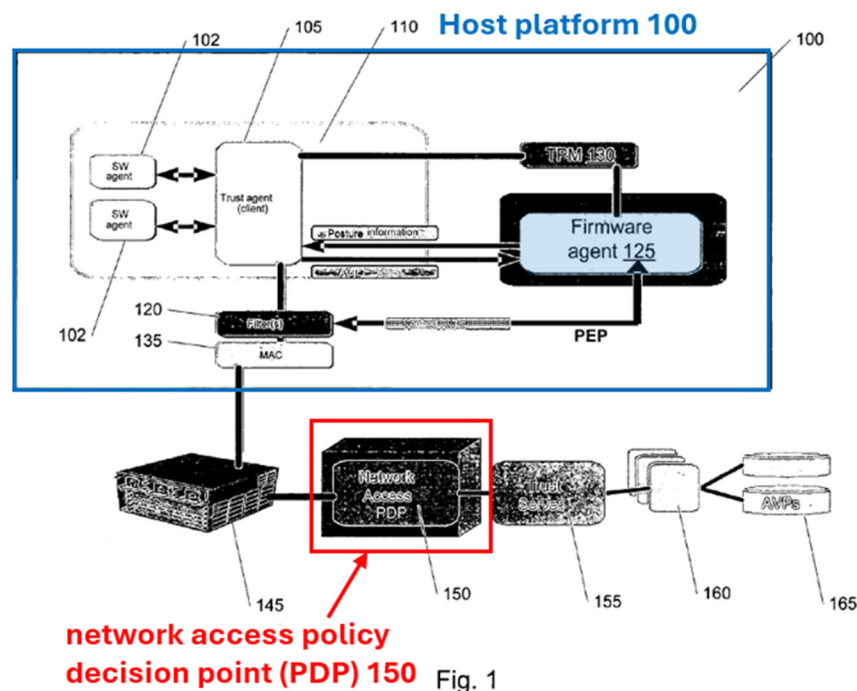
transmission feature, so as to ensure integrity of the activity data transmitted to server utility 150.” *Id.*, [0057], [0062]. And Bittmann does not limit its server to the device’s own reports: mediation system 210 receives “other reports of activity” and correlates “reports from multiple sources,” and the rated activity can include “data activity reported by system resources (for example probes, switches, VoIP servers, etc).” *Id.*, [0042]–[0043]. In my opinion, these teachings would have logically commended themselves to a POSITA considering how to verify that a device-side component implementing a service policy is accurately reporting usage.

C. Overview of Durham

77. U.S. Patent Application Publication No. 2007/0006282 A1 to Durham et al. (“Durham”) was filed June 30, 2005 and published January 4, 2007, before the assumed January 28, 2009 priority date of the ’777 Patent. Durham, cover.

78. Durham addresses how a network can assess a host and enforce access rules “even when the host operating system is compromised.” Durham, [0010]. Durham describes “[a] service processor, labeled firmware agent 125,” which “may be used for tamper resistant posture reporting and remediation of host platform 100.” Durham, [0012]. Ordinary software agents 102 and trust agent 105 operate within the host software environment, while firmware agent 125 operates separately in an environment that host-operating-system code cannot tamper with, disable, or circumvent. *Id.* Specifically, Durham explains that “no code executing within host

operating system 110 can tamper with firmware agent code, prevent firmware agent code from running, or circumvent operation of the firmware agent,” and that firmware agent 125 “has exclusive access to host resources including, for example, traffic filters 120 on a network interface card.” *Id.* Durham thus explains that firmware agent 125 “may serve as a tamper-resistant execution environment on host platform 100” and may host code that attests to the integrity of the host’s software. Durham, [0013].



Durham, Fig. 1.

79. Durham's Figure 1 shows how host platform 100 obtains network access. In a network access control exchange, trust agent 105 initiates a connection to network access server 145, "communicating the intent to connect to the network," and network access server 145 offloads the access decision to network access policy decision point ("PDP") 150. Durham, [0015]. Until the network decides, firmware agent 125 "may prevent any other traffic from entering or exiting host platform 100 other than the network access control channel." *Id.* That is Durham's default configuration: "by default filters 120 block all access by host system 100, but firmware agent 125 may enable only network access control exchanges." *Id.*, [0021]. Responses from PDP 150 then "either allow or prevent access based, at least in part, on evaluation of credentials and posture information conveyed from host platform 100." *Id.*, [0016].

80. Durham ties firmware agent 125 to a particular host platform through a secure enrollment procedure that may "cryptographically bind the identities of host platform 100 with firmware agent 125"; those identities may be "protected using Trusted Platform Module (TPM) 130." Durham, [0017]. Enrollment establishes "unique identities for host platform 100 and for firmware agent 125, for example, using Public-Key/Private-Key Identity Certificates," and creates an attribute certificate that binds the two, resulting in "an established authenticated binding between the host environment(s) and firmware agent 125." *Id.*, [0019]. When the

host seeks access, PDP 150 sends “an unpredictable (random) nonce session identifier” to “establish proof-of-liveness for firmware agent 125.” *Id.*, [0027]. Firmware agent 125 returns posture measurements as attribute-value pairs “signed with a private/symmetric key of firmware agent 125,” and it includes the PDP’s nonce in a digital-signature attribute-value pair that it sends to PDP 150. *Id.*, [0025], [0028]. A POSITA would have understood that, where firmware agent 125’s identity is a Public-Key/Private-Key Identity Certificate, its response, which includes the PDP’s nonce, is signed with the private key that corresponds to that enrolled identity. *Id.*, [0019], [0025], [0028].

81. On the network side, trust server 155 evaluates the posture information that firmware agent 125 sends to PDP 150 “first for integrity and authenticity of the data, using digital signature verification,” and then compares the posture attribute-value pairs with “administrative policies 160” to determine whether to allow host platform 100 to “connect to additional network resources.” Durham, [0029]. The PDP’s evaluation “may include validity and proof of liveness.” *Id.*, [0043]. Because the signature covers the PDP’s nonce, a valid signature shows that firmware agent 125 produced the response for the current exchange rather than replaying an earlier one. *Id.*, [0025], [0027]–[0028]. Durham also explains that “if a rogue trust agent modifies or does not send firmware agent 125 integrity measurements, this action

can be detected at the server ... and appropriate remedial action can be taken.” *Id.*, [0026].

82. The network then directs the host’s access. PDP 150 returns a signed result whose attribute-value pairs “can provide firmware agent 125 with functionality parameters including, for example, access control lists, filters, to be installed by firmware agent 125 and/or instructions on how to fix or remediate host platform 100.” Durham, [0030]. If the connection passes, PDP 150 “can return Layer 3 or Layer 4 traffic rules to host platform 100, which may be applied with a high degree of assurance via firmware agent 125.” *Id.*, [0032]. If “access to the network is denied,” the returned attribute-value pairs “may contain instructions for remediation,” and PDP 150 “may return access control lists to set filters 120.” *Id.*, [0031]. The host passes the returned policy to firmware agent 125 through a control channel, and firmware agent 125 verifies the signature and nonces before installing it; “if verification is not successful, the firmware agent may install a default ACL or take other default remediation steps.” *Id.*, [0046].

83. As explained in the Field of Endeavor section above, the field of endeavor of the ’777 Patent is managing an end-user device’s access to and use of network services. Durham is in that field. Durham relates to “techniques for enforcement of network access policies” for networked electronic devices. Durham, [0001]. Network access PDP 150 evaluates the credentials and posture information

conveyed from host platform 100 and allows or prevents the host's network access, and firmware agent 125 enforces that decision on the host as a "trusted policy enforcement point (PEP)" that enforces policies "even when the host operating system is compromised." *Id.*, [0010], [0015]–[0016], [0029]–[0032]. Wright and Bittmann likewise concern managing a device's access to and use of network services, as explained in the Overview of Wright and Overview of Bittmann sections. Accordingly, Durham is in the same field of endeavor as the '777 Patent, Wright, and Bittmann.

84. Durham is also reasonably pertinent to a problem addressed by the '777 Patent: verifying that device-side components implementing a service policy are present, properly configured, and accurately reporting usage. As explained in the Problem Solved by the Inventors of the '777 Patent section above, the '777 Patent describes a service controller that authenticates the device's service processor to verify that it is present and properly configured to implement the device portion of a service policy, and that checks whether the service processor reports the device's usage in the expected manner. '777 Patent, 48:9–17, 48:65–49:19, 58:3–11. Durham addresses the same problem for the device-side agent that reports a host's condition to the network and enforces the network's access policy: it seeks to enforce network access policies "even when the host operating system is compromised." Durham, [0001], [0010]. To that end, the network sends firmware agent 125 a fresh nonce to

“establish proof-of-liveness,” verifies the agent’s signed response to confirm that it came from the enrolled agent and was not modified, and evaluates the agent’s integrity attestations about the code and components running on the host, “that they are the correct image, uncompromised, and have not been circumvented.” *Id.*, [0013], [0025]–[0029]. Durham also detects at the server when “a rogue trust agent modifies or does not send firmware agent 125 integrity measurements.” *Id.*, [0026]. These teachings would have commended themselves to a POSITA considering how a network can verify that a device’s service processor is present, properly configured, and accurately reporting.

VII. SUMMARY OF UNPATENTABILITY

85. I have reproduced the Proposed Grounds of Unpatentability from the Petition for ease of reference:

Ground (all §103)	Claims	Prior Art
1A	1–7, 10, 11, 13, 14	Wright
1B	15–17, 19–22, 28, 29, 31, 32	Wright-Bittmann
1C	8, 9, 12, 23–25	Wright-Durham
1D	16, 18, 23, 26, 27, 30, 33	Wright-Bittmann-Durham

VIII. GROUNDS

A. GROUND 1A: CLAIMS 1–7, 10, 11, 13, AND 14 ARE OBVIOUS OVER WRIGHT

86. My overview of Wright above describes system 200, client 201, and the exchanges between them on which the following analysis relies. For the reasons below, in my opinion Wright renders obvious claims 1–7, 10, 11, 13, and 14.

1. Claim 1

a. 1[pre]: “A method performed by a network system, the method comprising:”

87. In my opinion, Wright discloses or suggests 1[pre]. Wright’s system 200 “allows an administrator to manage and distribute policies and software upgrades, analyze logs, and perform remote diagnostics,” and it may execute on a standalone computer or “be implemented by a series of networked computers as may typically be implemented by an enterprise” (*network system*). Wright, [0046]. Wright explains that Figures 2A and 2B “illustrate interaction between a computer system 200 (FIG. 2A) acting in a server role with respect to a mobile computer system 201 (FIG. 2B) acting in a client role for the purposes of managing security with respect to the client system.” *Id.*

88. System 200 performs the method I describe for 1[a]–1[d]. It receives the encrypted token, which, as I explain for 1[a], a POSITA would have understood to be authorization information accompanying the status request; obtains the policy

applicable to that device; receives the SHA1 response that client 201 computes in Establish Session; and uses that response to determine whether client 201 is authenticated. Wright, [0078], [0104], [0128]–[0129], [0133]–[0147]. Client 201, by contrast, is the software on the mobile device that takes part in those exchanges by sending the token and the SHA1 response; system 200 performs the receiving, obtaining, and authentication operations that claim 1 recites. *Id.*, [0046].

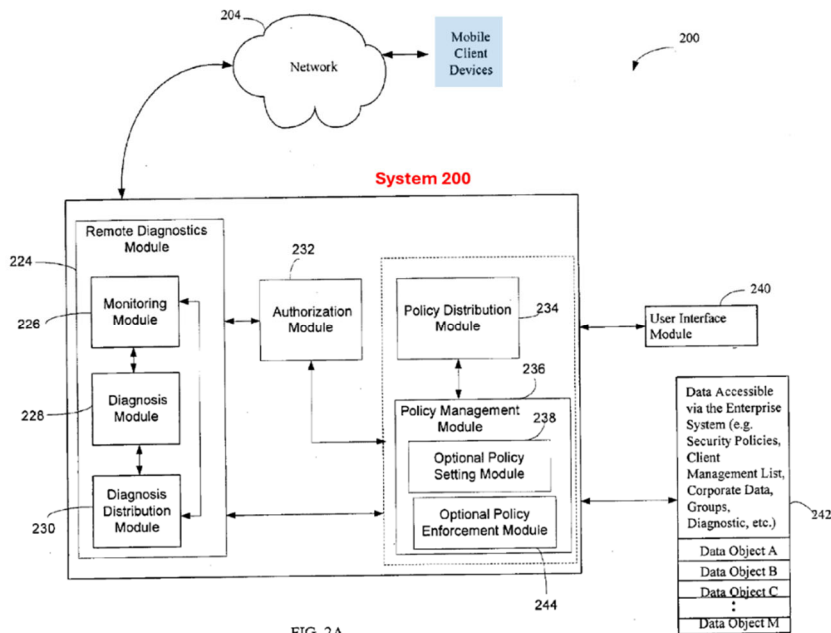


FIG. 2A

Wright, Fig. 2A.

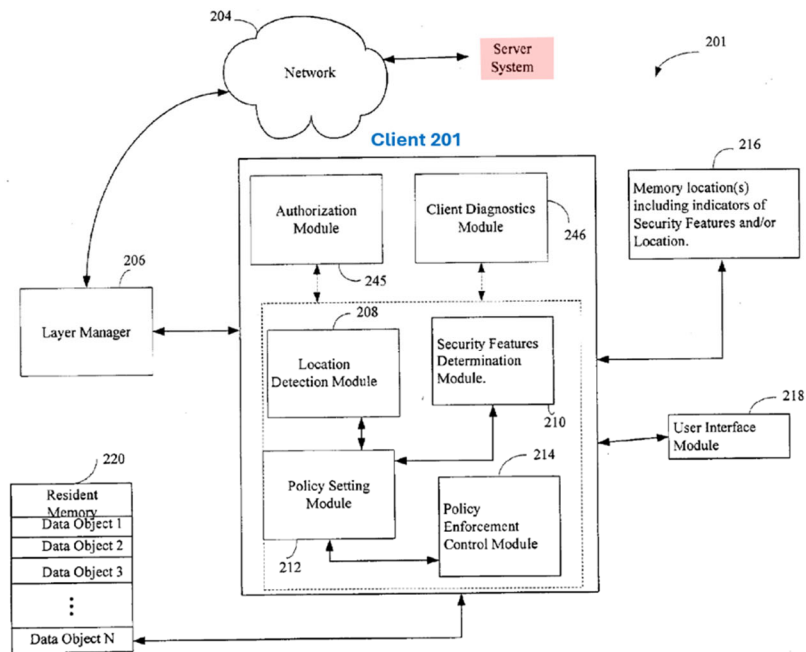


FIG. 2B

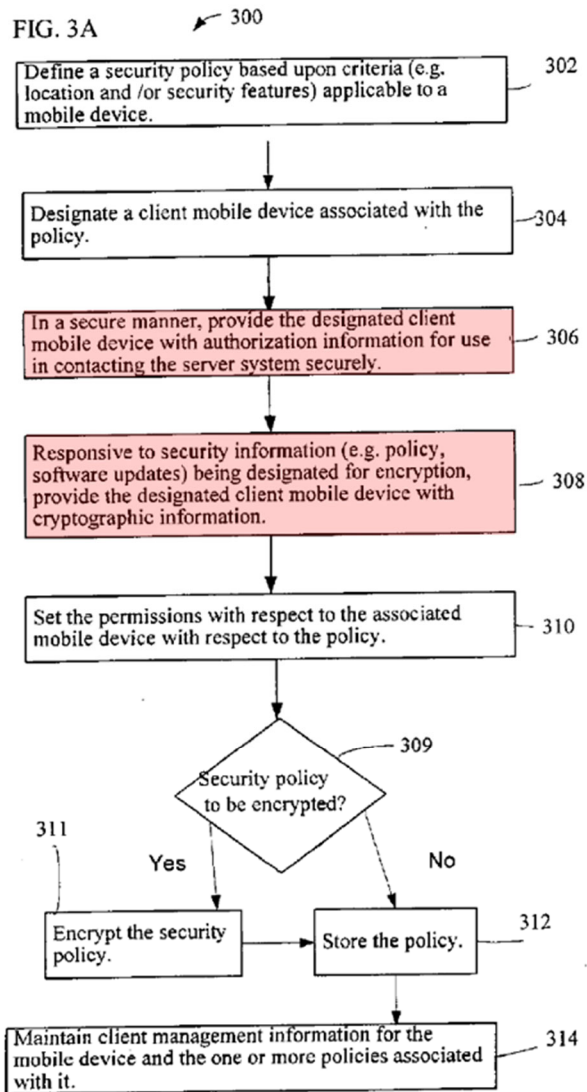
Wright, Fig. 2B.

89. Accordingly, Wright discloses or suggests that system 200, which may be implemented “by a series of networked computers” (*network system*), performs the method described for 1[a]–1[d] (*method performed by a network system*). Wright, [0046].

- b. **1[a]:“receiving a device credential from an end-user device communicatively coupled to the network system over a wireless access network;”**

90. In my opinion, Wright discloses or suggests 1[a]. During set-up, system 200’s policy management module 236 defines a security policy applicable to a client mobile device (step 302 of Figure 3A) and “designates 304 one or more client devices associated with the policy.” Wright, [0078]. “In a secure manner,” module

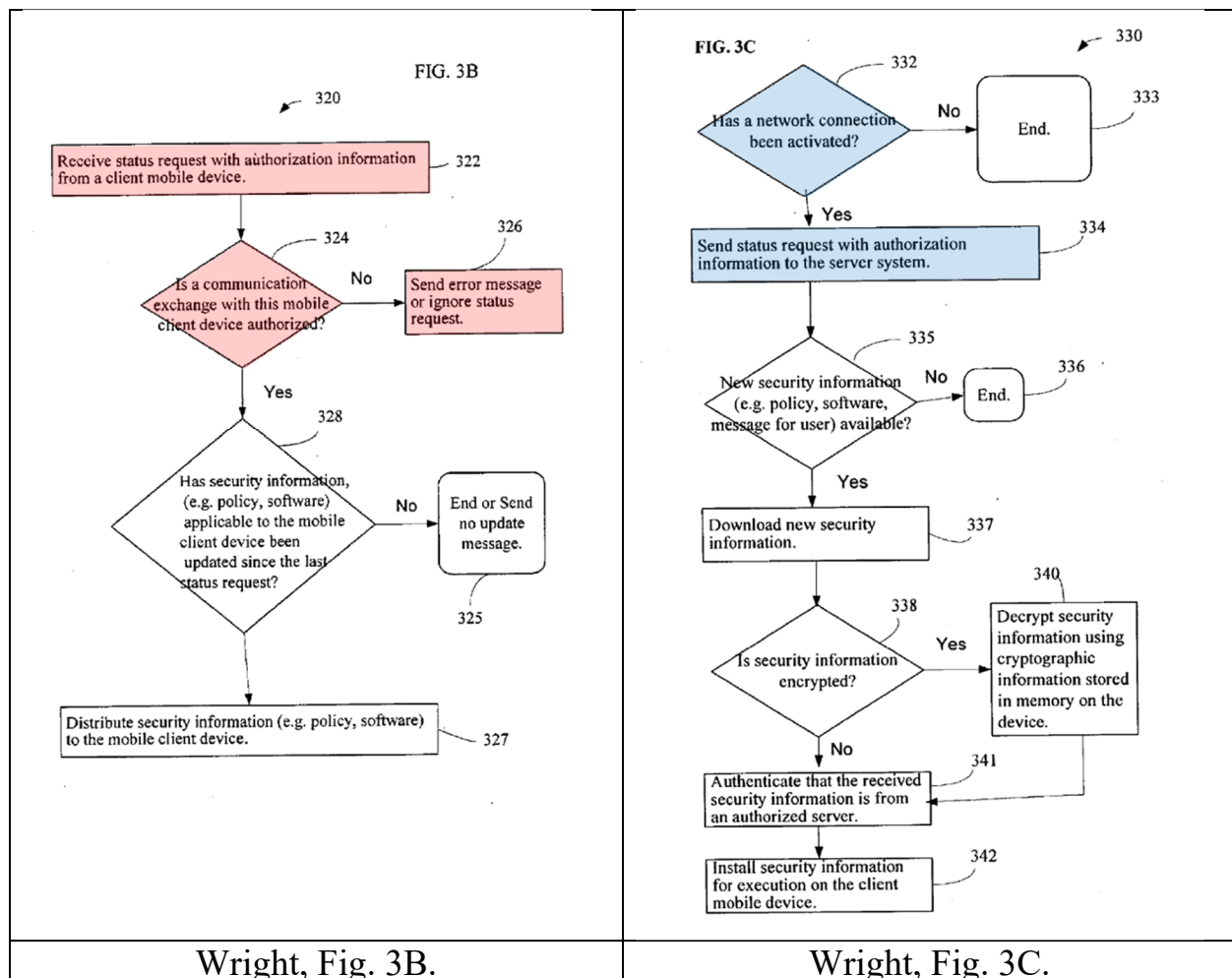
236 then provides (step 306) each designated client mobile device with “authorization information for use in contacting the server system securely,” and Wright’s example is “an encrypted token provided by the authorization module 232 to the mobile client device during a trusted connection between the two.” *Id.* Module 236 separately provides (step 308) cryptographic information, such as “a key for use with a cryptographic authentication protocol.” *Id.* That key is not unique to a client. It is the key that Wright generates when system 200 is installed and uses for “authentication of clients and authentication of a downloaded security information such as policies by a client”: the client decrypts a downloaded policy “using cryptographic information stored in memory (e.g. 220) that was previously downloaded in a secure session within the enterprise’s internal network,” and the policy file is “encrypted with the key.” *Id.*, [0130]–[0131]. Wright stores that key in an encrypted file, and “[t]his encrypted file is copied and distributed to the clients and any other servers in the server system”; “[c]lients installed by some other enterprise will have a different key, and clients installed in an unmanaged mode will not have a key.” *Id.*, [0130]. A POSITA would have understood from this description that every client the enterprise installs holds the same key, so the key distinguishes the enterprise’s managed clients from other clients but does not distinguish one managed client from another. *Id.*



Wright, Fig. 3A.

91. The mobile device later presents that authorization information to system 200. Each client “sends a status request upon network access starting up to the server system,” and “[t]he status request provides information regarding the client mobile device, examples of which are authorization information, policies and

their version numbers resident on the device, and diagnostic information.” Wright, [0128]. As Figure 3C shows, once policy setting module 212 determines that a network connection has been activated, client 201’s authorization module 245 sends the status request “along with authorization information to the server system 200” (step 334). *Id.*, [0129]. As Figure 3B shows, system 200’s authorization module 232 “receives 322 the status request with authorization information from a client mobile device, and it determines 324 whether a communication exchange with this mobile device is authorized.” *Id.*, [0128].



92. Wright's description of Figure 3B explains that system 200's authorization module 232 decides whether to authorize client 201's exchange (step 324), and Wright explains that the enterprise key is used for "authentication of clients." Wright, [0128], [0130], Fig. 3B. Wright also explains that authorization module 232 may perform that authorization with "[v]arious authorization protocols and techniques," including "a cryptographic authentication protocol." *Id.*, [0052]. The description of Figures 3B and 3C, however, describes the authorization decision only functionally: it states that authorization module 232 "determines 324 whether a communication exchange with this mobile device is authorized," but it does not provide implementation details for a cryptographic authentication protocol, such as what messages client 201 and system 200 exchange or what system 200 computes. *Id.*, [0128]–[0129].

93. Wright's Establish Session protocol is such a cryptographic authentication protocol. Wright, [0133]. Wright describes it as a protocol "that may be used to authenticate the client as a first step in all client-server transactions." *Id.* In Establish Session, client 201 sends its protocol version and user identification (UID), system 200 sends a random challenge, and client 201 returns an SHA1 hash computed from the challenge and its key. *Id.*, [0134]–[0136]. System 200 computes the same hash and compares it with client 201's response; if they do not match,

system 200 “drops the connection with no further ado,” and if they match, system 200 has authenticated client 201. *Id.*, [0137]. As I explained in my overview of Wright, the protocol is cryptographic because it uses SHA1, a cryptographic hash, keyed with the enterprise key that Wright distributes to its managed clients for “authentication of clients,” so that only a client holding the key can produce the matching response. *Id.*, [0130], [0136]–[0137].

94. In my opinion, a POSITA would have understood Wright to disclose or suggest that Establish Session is one way for system 200 to perform the authorization/authentication of client 201 at steps 322–326 of Figure 3B when client 201 sends its status request at step 334 of Figure 3C. Wright, [0128]–[0129], [0133]–[0138]. Several features of Wright’s description lead a POSITA to that understanding. Both descriptions concern the same system 200 and client 201: Establish Session is the protocol by which “a session may be established between the server system 200 and the client system 201.” *Id.*, [0133]. Wright states that authorization module 232 may use “a cryptographic authentication protocol,” and Establish Session is the cryptographic authentication protocol Wright describes for authenticating clients with the enterprise key. *Id.*, [0052], [0130], [0133]–[0137]. Wright states that Establish Session “may be used to authenticate the client as a first step in all client-server transactions,” and the status-request exchange of Figures 3B and 3C is such a transaction, in which, once system 200 receives the status request

at step 322, the first thing it does is decide authorization at step 324. *Id.*, [0128], [0133]. And both descriptions lead to the same next steps: after a successful authorization in Figure 3B, system 200 determines whether updated security information applies to the device and distributes it, and after Establish Session, client 201 queries for and downloads updated policies. *Id.*, [0128], [0138]–[0147].

95. Establish Session both authorizes and authenticates client 201. It authenticates client 201 because a matching response shows that client 201 holds the enterprise key, at which point “the server has authenticated the client.” Wright, [0137]. It also authorizes the exchange, because system 200 continues the session only for a client that passes and otherwise ends it; Wright accordingly refers to “a session not being authorized or being established.” *Id.*, [0137]–[0138]. On failure, system 200 drops the connection, which corresponds to step 326 of Figure 3B, where an unauthorized status request “is ignored 326 or an error message is sent 326”: in either case system 200 does not carry the exchange forward. *Id.*, [0128], [0137]–[0138]. On success, the steps proceed as Figures 3B and 3C show. Wright describes the steps that follow Establish Session as proceeding “[a]s in FIG. 3C”: when new security information is available, it is downloaded. *Id.*, [0138]–[0141], [0144].

96. Even if Establish Session were viewed as a separate “alternate embodiment,” in my opinion a POSITA would have been motivated to use Establish Session to perform the authorization/authentication of Figures 3B and 3C. Wright,

[0133]. First, because Figures 3B and 3C do not specify how authorization module 232 authorizes or authenticates client 201, a POSITA implementing them would have looked to Wright's Establish Session disclosure for the cryptographic implementation detail. *Id.*, [0128]–[0129], [0133]–[0137]. Wright itself points there: it identifies “a cryptographic authentication protocol” as a technique for authorization module 232, and Establish Session is the cryptographic authentication protocol that Wright describes for the same system 200, client 201, and enterprise key. *Id.*, [0052], [0130], [0133]. Second, Wright describes authorization module 232 as “a further safeguard against unauthorized or rogue mobile devices trying to hijack the security policies or corporate data.” *Id.*, [0052]. A POSITA would have been motivated to authorize and authenticate only clients that pass a cryptographic authentication protocol, because a keyed challenge-response check gives that safeguard force: a rogue device without the enterprise key cannot produce the matching response to a fresh random challenge. *Id.*, [0130], [0135]–[0137].

97. A POSITA would have had a reasonable expectation of success in using Establish Session to perform that authorization/authentication. Wright describes each protocol step: the messages client 201 and system 200 exchange, the value client 201 computes, the comparison system 200 makes, and the result of a match or a mismatch. Wright, [0133]–[0137]. Wright also explains how the steps that follow authorization/authentication proceed “[a]s in FIG. 3C.” *Id.*, [0144]. And every input

the protocol needs is already in place: system 200 generates the key at installation and distributes it to client 201, and client 201 already uses that key and computes SHA1 hashes when it decrypts and authenticates downloaded policies. *Id.*, [0130]–[0132].

98. In my opinion, a POSITA also would have understood that, when system 200 performs the authorization/authentication with the Establish Session protocol, it would receive “authorization information” that includes the client device’s encrypted token. Establish Session performs the authorization of steps 322–326, which system 200 makes on the status request client 201 sends at step 334, and that status request carries the device’s “authorization information.” Wright, [0128]–[0129]. Wright’s example of authorization information is the encrypted token, which system 200 provisioned to the device as “authorization information for use in contacting the server system securely” and “provided ... during a trusted connection between the two.” *Id.*, [0078]. The status request at steps 322 (Figure 3B) and 334 (Figure 3C) expressly includes such “authorization information.” *Id.*, [0128]–[0129]. A client provisioned with the token would therefore present it when it seeks the authorized exchange that Establish Session begins.

99. System 200 needs to identify the device, not only the user, because it applies device-level policies and maintains device-level records. Wright’s policies may apply to users or to “the mobile devices themselves,” and a group member may

be “the mobile device itself.” Wright, [0049], [0078]. Wright’s highest-priority group is device-based: a “stolen mobile device” group “always has the highest priority, because it doesn’t matter what other groups the device is associated with if the device is in the hands of a thief.” *Id.*, [0050]. And policy management module 236 “maintains 314 client management information for the mobile device and the one or more policies associated with it.” *Id.*, [0104]. In my opinion, a POSITA would have understood that, to apply the stolen-device group’s policy to a stolen device, or to keep the record for a particular device current, system 200 needs a device identifier that tells it which device sent a request, and therefore which device’s policy and record apply.

100. In my opinion, a POSITA would have understood the encrypted token to be the only per-device information that system 200 provides to client 201 and that client 201 provides during the exchange (*device credential*). At set-up, system 200 provides each designated device with two items that it later uses in its exchanges with system 200: the encrypted token, which authorization module 232 provides “to the mobile client device during a trusted connection between the two,” and the enterprise key, which is copied to every client. Wright, [0078], [0130]. Only the token is issued to a particular device. During the exchange, client 201 sends the status request with its authorization information (the token), the versions of its resident policies, and diagnostic information, and, in Establish Session, its protocol

version, its user identification, and the SHA1 response. *Id.*, [0128], [0134]–[0136]. As I explain next, none of those three items from Establish Session identifies the particular client device.

101. The UID identifies the user, and a user can have several devices. Wright, [0134]. Wright’s first Establish Session message lists it as “(user identification) UID.” *Id.* Wright’s “Home” example shows two mobile devices, laptop 148 and PDA 150, each communicating “via a network connection at a user’s home” through the same wireless network access point. *Id.*, [0042], Fig. 1. In my opinion, a POSITA would have understood from that example that one user commonly has more than one mobile device, such as a notebook computer and a PDA. A UID received from one of a user’s devices would not tell system 200 which of the user’s devices sent it. In my opinion, a POSITA would therefore have understood that the UID does not allow system 200 to identify the device.

102. The SHA1 response does not identify the device either. Client 201 computes “SHA1(random challenge, key)” using the key common to all of the enterprise’s clients. Wright, [0130], [0136]. Any of the enterprise’s managed clients would compute the same response to the same challenge, so a matching response shows only that the device is one of those managed clients, not which one. *Id.* Wright’s statement that the enterprise-side servers outside the firewall “perform key/identity management” and communicate with the authority server for “client

management information such as a key, a user, a group, and version information associated with a client mobile device system” does not change my opinion. *Id.*, [0078]. That passage describes information those servers obtain about a client; it does not describe generating a different key for each client. Wright’s description of the key itself is that one key is generated when system 200 is installed, that the same encrypted file is copied to the clients, and that clients installed by another enterprise have a different key. *Id.*, [0130]. A key associated with a client in those servers’ records is consistent with that description, because each managed client holds the enterprise key.

103. Nor do the policy versions and diagnostic information that may be included in the status request identify a device. Wright, [0128]. The policy versions describe which versions of the policies are resident on the device, and many devices assigned the same policy hold the same versions. The diagnostic information describes the device’s condition, such as events, audit logs, and test results, rather than which device it is. *Id.*, [0054], [0128].

104. Nor would the network address from which a status request arrives identify the device to system 200. Wright’s devices move among the “Home,” “Work,” “Mobile,” and “Alternate” locations, connect through different access points, including one “associated with a cellular base station,” and reach system 200 over network 204 (e.g., the Internet). Wright, [0042]–[0045], [0047], [0128]. A

POSITA would have known that the Internet protocol (IP) address a mobile device uses is assigned by whichever network the device is using at the time, not by system 200, and that it changes as the device moves from one network to another. Wright itself treats the address that way: among its “criteria to be used for identifying a location,” Wright lists that “[t]he local IP address must match the stored value.” *Id.*, [0202], [0206]. The address therefore tells system 200, at most, where the traffic of a current connection comes from; it does not identify any of the enterprise’s devices. In my opinion, a POSITA would not have relied on the network address to identify a device to system 200; the token is what identifies to system 200 the device to which system 200 issued it.

105. Regardless, in my opinion a POSITA would have been motivated to have client 201 send a per-device token in the authorization/authentication exchange so that system 200 would be able to identify device-level policies, such as those of the highest-priority “stolen mobile device” group. Wright, [0050]. As I explained above, system 200 cannot identify the device from the UID, the SHA1 response, the policy versions and diagnostic information, or the network address from which the request arrives. Without a per-device value, system 200 would have no way to apply the stolen-device policy to the stolen device alone, rather than to every device of the same user or to none. Wright’s stolen-device group has the highest priority precisely because “it doesn’t matter what other groups the device is associated with if the

device is in the hands of a thief.” *Id.* The token that system 200 issued to the device during a trusted connection is the value that lets system 200 make that distinction. *Id.*, [0078].

106. A POSITA would have had a reasonable expectation of success in having client 201 send the token in that exchange. The status request already carries “authorization information.” Wright, [0128]–[0129]. Wright describes authorization information as including the encrypted token, which client 201 already holds because system 200 issued it at set-up. *Id.*, [0078]. And Wright’s session protocols “are designed to be extensible and straightforward to allow addition of new verbs later,” so carrying the token in the exchange would not have required any change to how system 200 issues the random challenge or verifies the SHA1 response. *Id.*, [0133]–[0137].

107. Client 201 sends the status request “upon network access starting up,” that is, once policy setting module 212 determines that a network connection has been activated. Wright, [0128]; *see also id.*, [0129]. System 200 communicates with the devices over network 204 (e.g., the Internet). *Id.*, [0047], [0128]. The mobile devices obtain that network access through wireless network access points (WAPs), including Mobile WAP 130 (an 802.11b connection at an airport), Home WAP 146, Work WAP 162, and Alternate WAP 182, which “is associated with a cellular base station providing network access through a General Packet Radio Services (GPRS)

system, Global System for Mobile communication (GSM) system, third generation wireless 3G system or other kind of mobile wireless communication system.” *Id.*, [0042]–[0045], Fig. 1. Each such access point, with the wireless network it provides, gives the mobile device a wireless link to network 204 and through it to system 200. In my opinion, Wright therefore discloses that the devices communicate with system 200 through wireless access networks.

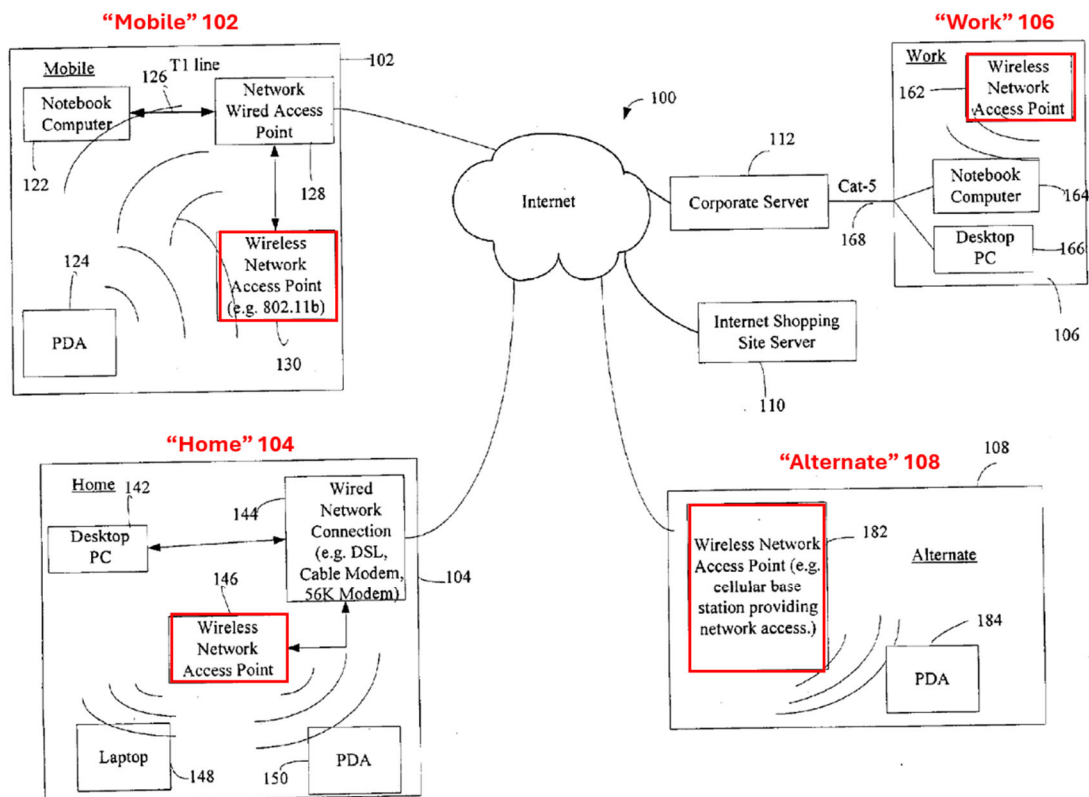


FIG. 1

Wright, Fig. 1.

108. Accordingly, Wright discloses or suggests that system 200 receives an encrypted token (*device credential*) from a client device (*end-user device*) that is coupled to system 200 through a wireless access network (e.g., WAPs 130/146/162/182) (*communicatively coupled to the network system over a wireless access network*). Wright, [0042]–[0045], [0078], [0128]–[0129], [0133]–[0137], Fig. 1.

c. 1[b]: “obtaining a service policy based at least in part on the device credential, the service policy being applicable to the end-user device;”

109. In my opinion, Wright discloses or suggests 1[b]. System 200’s policy management module 236 assigns policies to client devices and keeps a client management record for each device that includes its policy. Module 236 “designates 304 one or more client devices associated with the policy,” and it “maintains 314 client management information for the mobile device and the one or more policies associated with it.” Wright, [0078], [0104]. That information is stored in a data object accessible to system 200, and the administrator manages it for each client device. *Id.*, [0104], [0120]–[0121], [0126].

110. In my opinion, Wright’s *service policy* for a device has two portions, which correspond to the first and second portions recited in claims 5, 6, and 13. When system 200 evaluates the authorization information a device presents, Wright describes two outcomes, and each is governed by rules that system 200 applies to

that device. The first portion is the device's security rules, which system 200 sends to an authorized and authenticated device and which client 201 installs and enforces. Wright, [0048], [0128], [0132], [0145]–[0147]. The second portion is the denial of access that system 200 applies when authorization/authentication fails. *Id.*, [0128], [0137]. Together they define how system 200 treats the device: the rules the device enforces when it is admitted, and the denial it receives when it is not.

111. The first portion is the device's security rules. Those rules include the policy's "rules and permissions," the ports, network services, and applications to be monitored, and the "enforcement mechanisms to be put in place for a particular policy." Wright, [0048]. After successful authorization/authentication, system 200 sends any updated rules: in Figure 3B, policy distribution module 234 "distributes 327 the updated security information to the mobile device," and, after Establish Session, system 200 answers the "Download Policy" verb with "the encrypted policy file." *Id.*, [0128], [0145]–[0147]. Client 201 then authenticates and installs the rules, and policy enforcement control module 214 enforces them. *Id.*, [0064], [0068], [0132]. If no update is available, "the status request is ignored 325 or a no update message is sent 325." *Id.*, [0128]. In my opinion, a POSITA would have understood that, in that case, client 201 keeps enforcing the policy it already has: module 214 "comprises instructions for enforcing the security policy currently set by the policy setting module 212," and a no-update response leaves that policy in place. *Id.*,

[0064], [0128]. Either way, the device's security rules govern the device after it is authorized and authenticated.

112. The second portion denies the device access when authorization/authentication fails. Wright, [0128], [0137]. In Figure 3B, an unauthorized status request “is ignored 326 or an error message is sent 326,” and in Establish Session, system 200 “drops the connection with no further ado.” *Id.* That the second portion may be common to many devices does not change my opinion. As I read claim 1 under its ordinary meaning, the *service policy* must be *applicable to the end-user device*; claim 1 does not require that the policy be unique to that device. The denial applies to the device whenever that device fails authorization/authentication, so it is part of the policy applicable to that device. The '777 Patent likewise describes a second portion of a policy that suspends or denies a device's access: “the differences between the first network portion of a wireless access network service policy and the second network portion of a wireless access network service policy can include a difference in network access privileges, ... a quarantining of the device or user access capabilities, ... or suspending the device from access to the network.” '777 Patent, 51:9–28.

113. System 200 obtains the device's policy after successful authorization/authentication, based at least in part on the token. It determines the first portion of the device's policy, the security information “applicable to the mobile

client device,” using the device’s client management record. Wright, [0128]; *see also id.*, [0138]. In Figure 3B, policy distribution module 234 “determines 328 if security information (e.g. a policy or software) applicable to the mobile client device has been updated since the last status request.” *Id.*, [0128]. After Establish Session, client 201 sends a “Query Policy Update” message with its policy version, and system 200 answers “there is a newer version” or “your version is up-to-date.” *Id.*, [0138]–[0141]. In my opinion, a POSITA would have understood that, to make either determination, system 200 must look up which policy is assigned to the requesting device and the current version of that policy. That information is what module 236 keeps as “client management information for the mobile device and the one or more policies associated with it.” *Id.*, [0104], [0120].

114. To decide what is “applicable” to the device, system 200 must know which device sent the request, and the token is the only device identifier that was sent. As I explained for 1[a], system 200 issues the token to one device during a trusted connection, and client 201 on that device sends the token as authorization information with its status request. Wright, [0078], [0128]–[0129]. In my opinion, a POSITA would have understood that, in Wright’s design, the token that arrives with a request is how system 200 identifies which device sent it: system 200 identifies the sender as the device to which it issued that token. *Id.*, [0078], [0104], [0128]. None of the other information sent can be used to identify the device. The UID identifies

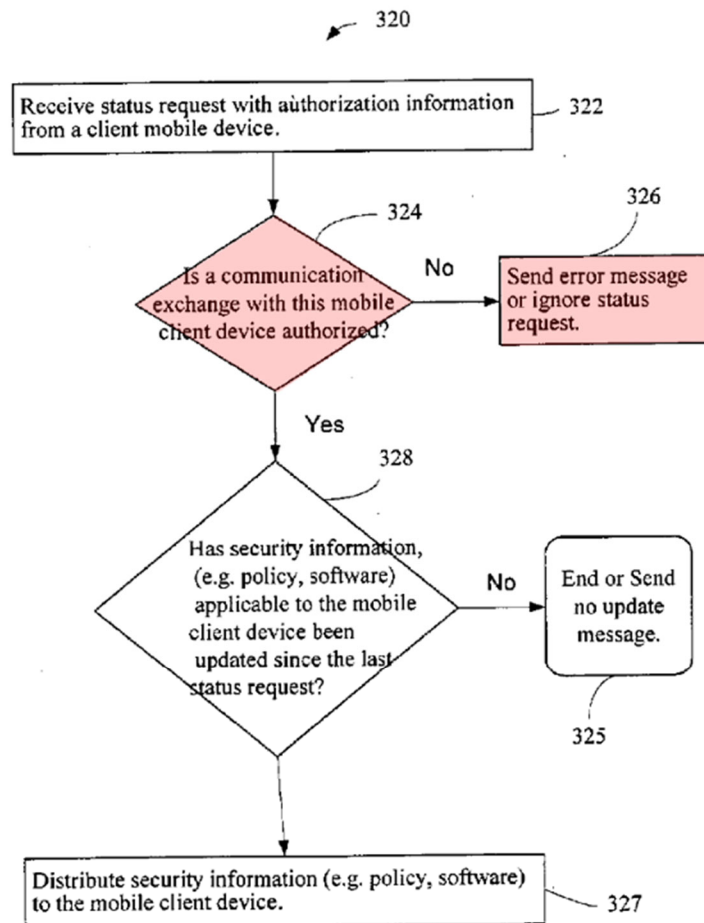
the user, who can have several devices; the SHA1 response is computed with a key common to all of the enterprise's clients; and the policy versions and diagnostic information describe the device's state, not its identity. *Id.*, [0042], [0128], [0130], [0134]–[0136]. The UID at most narrows the request to a user, so none of them can tell system 200 which device sent the request. A device running a copy of client 201 would present the same token, and because the token is the only device identifier in the request, system 200 would identify that device the same way, as the device to which it issued the token; that is the case Wright-Durham addresses with Durham's device-bound check, as I explain for 8[d] in Ground 1C.

115. In my opinion, a POSITA thus would have understood that the token identifies the first portion, the device's security rules. Because authorization module 232 issues the token to that device during a trusted connection, a POSITA would have understood that system 200 knows to which device it issued the token, and module 236 maintains "client management information for the mobile device and the one or more policies associated with it." Wright, [0078], [0104]. When the token arrives with the status request, system 200 uses it to identify the client management record, and therefore the security rules, of the device to which it issued that token. *Id.*, [0078], [0104], [0128]. Because the token identifies the first portion, system 200 *obtain[s] a service policy based at least in part on the device credential.*

116. In my opinion, system 200 also would obtain the second portion when authorization/authentication fails, because it cannot deny access without it. Wright, [0078], [0128]. System 200 provisions authorization information “for use in contacting the server system securely,” and it permits only authorized exchanges. *Id.*, [0052], [0078]. When step 324 fails, system 200 must apply the rule that governs an unauthorized device (ignore the request, send an error message, or drop the connection), and it can do so only by obtaining that rule for the device that failed. *Id.*, [0128], [0137].

117. System 200 would obtain the second portion *based at least in part on the device credential*. At step 324 of Figure 3B, system 200 decides authorization/authentication using the authorization information, including the token, that it received at step 322, and at step 326 it denies access when that decision fails. Wright, [0128]. As I explained for 1[a], the token is part of that authorization information. The token is therefore an input to the decision whose failure causes system 200 to obtain and apply the second portion.

FIG. 3B



Wright, Fig. 3B.

118. Wright obtains the policy after authorization/authentication completes: system 200 determines whether updated security rules apply to the device, and distributes them, only after the authorization decision at step 324 or a successful Establish Session. Wright, [0128], [0137]–[0147]. I understand that the steps of a method claim are not limited to the order in which they are written unless, as a matter

of logic or grammar, they must be performed in that order. Although claim 1 recites 1[b] before 1[c][i]–1[d], nothing in claim 1 requires obtaining the *service policy* before those steps: 1[c][i]–1[d] concern obtaining and using the service processor authentication information (in Wright, client 201’s SHA1 response), and that information does not depend on the *service policy*. The ’777 Patent itself states that “[i]n general, the order of the steps of disclosed processes may be altered within the scope of the invention.” ’777 Patent, 8:64–66.

119. In my opinion, a POSITA also would have found it obvious for system 200 to use the token to obtain the device’s complete policy, both the first and second portions, when the token arrives and before authorization/authentication completes. System 200 issues the token to one device and therefore knows the association between the token and that device. Wright, [0078]. It keeps a policy record for each device. *Id.*, [0104]. And it needs a way to look up device-level policies, because the “stolen mobile device” group always has the highest priority. *Id.*, [0050]. A POSITA would have recognized that indexing both portions of the policy by the token that identifies the device was the straightforward way to retrieve the device’s policy, because the token is the only device-level information the client provides, as I explained for 1[a].

120. Looking up both portions of the policy on arrival also would allow system 200 to apply whichever portion the outcome of authorization/authentication

calls for without delay, including the second portion on failure. The token arrives with the status request, while the outcome depends on information that arrives later: client 201's SHA1 response to system 200's random challenge. Wright, [0128]–[0129], [0134]–[0137]. A POSITA would have recognized that, by retrieving the policy while the challenge is outstanding, system 200 would have the security rules ready to send if the response matches, and the denial ready to apply if it does not.

121. In my opinion, looking up the policy before or after authorization/authentication was a choice between two predictable options, and a POSITA would have had a reasonable expectation of success with either. Either way, system 200 retrieves the same stored policy through the same stored association, and the timing of the lookup does not change the random challenge, the SHA1 computation, the comparison, or the consequences of a match or a mismatch. Wright, [0104], [0133]–[0137]. Looking up the policy early does not grant the client any access; access still depends on the outcome of authorization/authentication. *Id.*, [0128], [0137].

122. Implementing the lookup on arrival would merely have required storing the first and second portions of the *service policy* linked to the token and performing the lookup before authorization/authentication rather than after. For example, system 200 would store the security rules and the denial that apply to the device, linked to the token, with the client management information it already maintains for that

device. As I explained above, a POSITA would have understood that system 200 already knows to which device it issued the token. System 200 would then retrieve both portions using the token as soon as the token arrives, rather than after the challenge-response comparison. Wright, [0078], [0104], [0120]. Wright's existing modules would continue to define, distribute, and enforce the retrieved policy. *Id.*, [0047], [0061], [0068]. These would be changes to system 200's software. Wright explains that system 200 "may be implemented as software executing in a standalone computer having a processor being accessible to memory," and the linked portions would be stored with the client management information already held in memory 242. *Id.*, [0046], [0120]. No new hardware would be required.

123. Accordingly, Wright discloses or suggests *obtaining a service policy* comprising a first portion (security rules to be sent to and enforced by the device) and a second portion (denial of access) *based at least in part* on the encrypted token (*device credential*), where that policy is *applicable to the end-user device*. Wright, [0048], [0050], [0078], [0104], [0128], [0137]–[0147].

d. 1[c][i]:“obtaining service processor authentication information associated with a service processor on the end-user device,”

124. In my opinion, Wright discloses or suggests 1[c][i]. Wright's *service processor* is client 201, which "may be implemented as software executing in a mobile computing device" and "enforces the policies set up by the administrator."

Wright, [0046]. As Figure 2B shows, client 201 includes several software modules on the mobile device, including policy setting module 212, policy enforcement control module 214, and authorization module 245. *Id.*, [0058]. Client 201 is therefore a *service processor on the end-user device*.

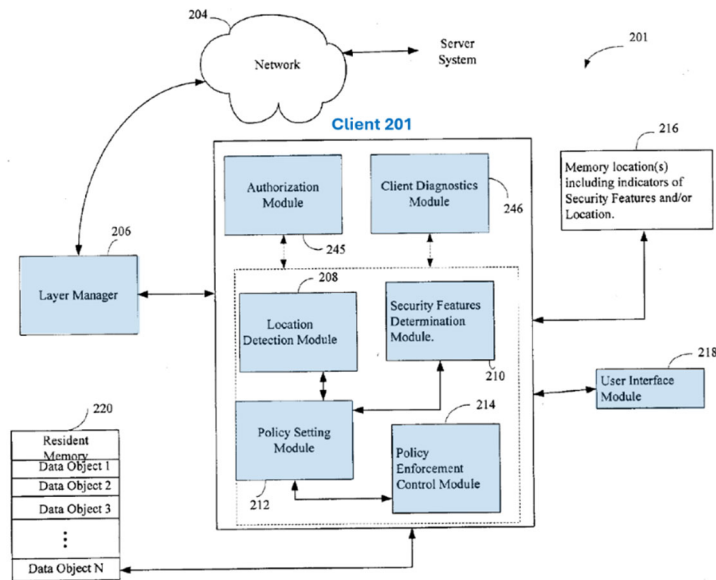


FIG. 2B

Wright, Fig. 2B.

125. As I explained for 1[a], Establish Session is one way for system 200 to perform the authorization/authentication of client 201 at steps 322–326 of Figure 3B when client 201 sends its status request at step 334 of Figure 3C. Wright, [0052], [0128]–[0129], [0133]–[0147]. In my opinion, it also would have been obvious to implement that authorization/authentication using Establish Session, because Figures 3B and 3C do not specify how the client is authorized or authenticated, authorization module 232 is a safeguard against rogue devices, and Wright describes

each protocol step and how the later steps proceed “[a]s in FIG. 3C.” *Id.*, [0052], [0133]–[0137], [0144].

126. Establish Session has three protocol steps. Client 201 first sends its protocol version and user identification (UID) (step 1). Wright, [0134]. System 200 responds with a “random challenge” (step 2). *Id.*, [0135]. When system 200 was installed, a key was generated and distributed to the enterprise’s clients for “authentication of clients.” *Id.*, [0130]. In step 3, client 201 returns “SHA1(random challenge, key),” an SHA1 hash computed on the mobile device from that challenge and the client’s key. *Id.*, [0136]. In my opinion, a POSITA would have understood that client 201’s authorization module 245 performs this step. Module 245 is the client module that “provides authorization information to the authorization module 232 of the server system 200 to establish communication exchanges with the client mobile device,” it sends the status request that begins the exchange, and it already uses the same key to authenticate downloaded policies. *Id.*, [0059], [0129]–[0132].

127. Client 201 can return the correct SHA1 hash only if it has the enterprise’s key. Wright, [0130]. The response is a cryptographic hash of the random challenge and the key, so a device without the key cannot compute the value that system 200 expects, and a device with another enterprise’s key computes a different value: “[c]lients installed by some other enterprise will have a different key, and clients installed in an unmanaged mode will not have a key.” *Id.*, [0130], [0136]–

[0137]. The key is thus a secret held by the enterprise's managed clients, distinct from the encrypted token that identifies the device for 1[a] and 1[b]. *Id.*, [0078], [0130].

128. System 200 computes the same SHA1 hash, using the random challenge it sent and the key, and compares it with client 201's response to determine whether the *service processor* is authorized and authenticated, as I explain for 1[d]. Wright, [0130], [0137]. The SHA1 response is therefore *service processor authentication information*: it is the information system 200 uses to authenticate client 201, because if the values match, "the server has authenticated the client." *Id.*, [0137]. And because client 201, executing on the mobile device, generates the response from the random challenge and its key, the response is *associated with a service processor on the end-user device*. *Id.*, [0130], [0135]–[0136].

129. Accordingly, Wright discloses or suggests system 200 receiving the SHA1 response (*obtaining service processor authentication information*). Further, Wright discloses or suggests that client 201 (*service processor*) computes that SHA1 hash on the mobile device using the random challenge and its key, and that client 201 sends it to system 200, e.g., via authorization module 245 (*service processor authentication information associated with a service processor on the end-user device*). Wright, [0059], [0129]–[0130], [0133]–[0137].

- e. 1[c][ii]: “the service processor comprising one or more device agents at least assisting to implement the service policy; and”**

130. In my opinion, Wright discloses or suggests 1[c][ii]. Client 201 (*service processor*) includes, among other modules, policy setting module 212, policy enforcement control module 214, and authorization module 245 (*one or more device agents*). Wright, [0058], Fig. 2B. Each of these modules is a software component of client 201 that runs on the mobile device and performs a defined task in the device's policy system, which is why a POSITA would have understood each to be a device agent. *Id.*, [0046], [0059], [0061], [0064], [0068].

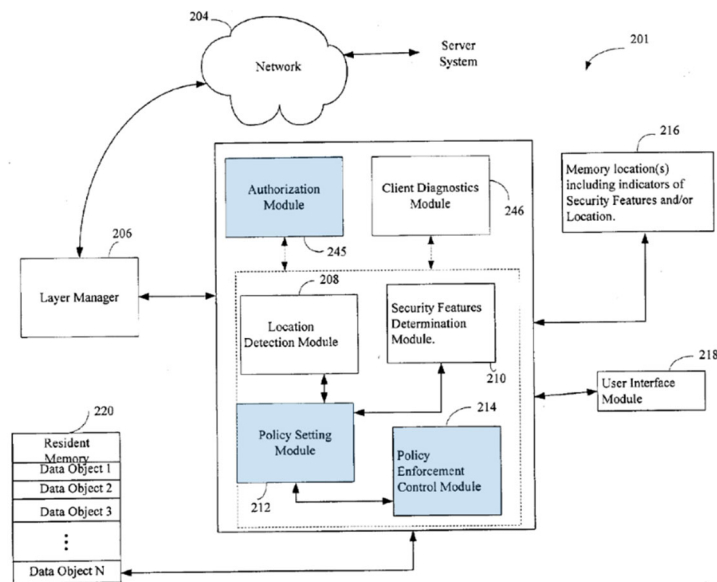


FIG. 2B

Wright, Fig. 2B.

131. These modules *at least assist[] to implement the service policy*, including both its first portion (security rules sent to and enforced by the device) and its second portion (denial of access), as I described for 1[b]. As to the first portion, policy setting module 212 “receives, installs and updates” the security policies that policy management module 236 provides through policy distribution module 234. Wright, [0061]. These are the policies “applicable to the requesting client device” that system 200 distributes. *Id.*, [0128], [0146]–[0147]. After a policy is downloaded, policy setting module 212 determines whether it is encrypted and, if so, decrypts it; authorization module 245 authenticates it; and the policy is “installed 342 for execution on the client mobile device.” *Id.*, [0131]–[0132]. Policy setting module 212 “communicates the current security policy to the policy enforcement control module 214,” and module 214 “comprises instructions for enforcing the security policy currently set by the policy setting module 212.” *Id.*, [0064], [0068]. Module 214 implements the policy through its “enforcement mechanism modules as specified by the policy”; for example, it may prevent certain files from being transferred over a wireless connection. *Id.*, [0068].

132. As to the second portion, authorization module 245 sends the status request “along with authorization information” to system 200. Wright, [0129]. In my opinion, a POSITA would have understood that this authorization information includes the encrypted token, as I explained for 1[a], and that module 245 also

returns the SHA1 response in Establish Session, as I explained for 1[c][i]. *Id.*, [0059], [0078], [0128]–[0129], [0136].

133. System 200 uses that information to determine whether client 201 is authorized and authenticated and, if not, “drops the connection with no further ado.” Wright, [0128], [0137]. By sending the information on which system 200 decides whether to deny access, module 245 *at least assist[s] to implement* the second portion of the *service policy*: without the token and the SHA1 response from module 245, system 200 would have nothing on which to decide whether the device is to be admitted or denied. *Id.*, [0059], [0128]–[0129], [0137].

134. Accordingly, Wright discloses or suggests that client 201 (*service processor*) includes modules 212, 214, and 245 that receive, authenticate, install, and enforce the first portion of the *service policy*, and that module 245 also provides the information to determine whether to execute the second portion of the *service policy* (*one or more device agents at least assisting to implement the service policy*). Wright, [0059], [0061], [0064], [0068], [0129], [0131]–[0132], [0137].

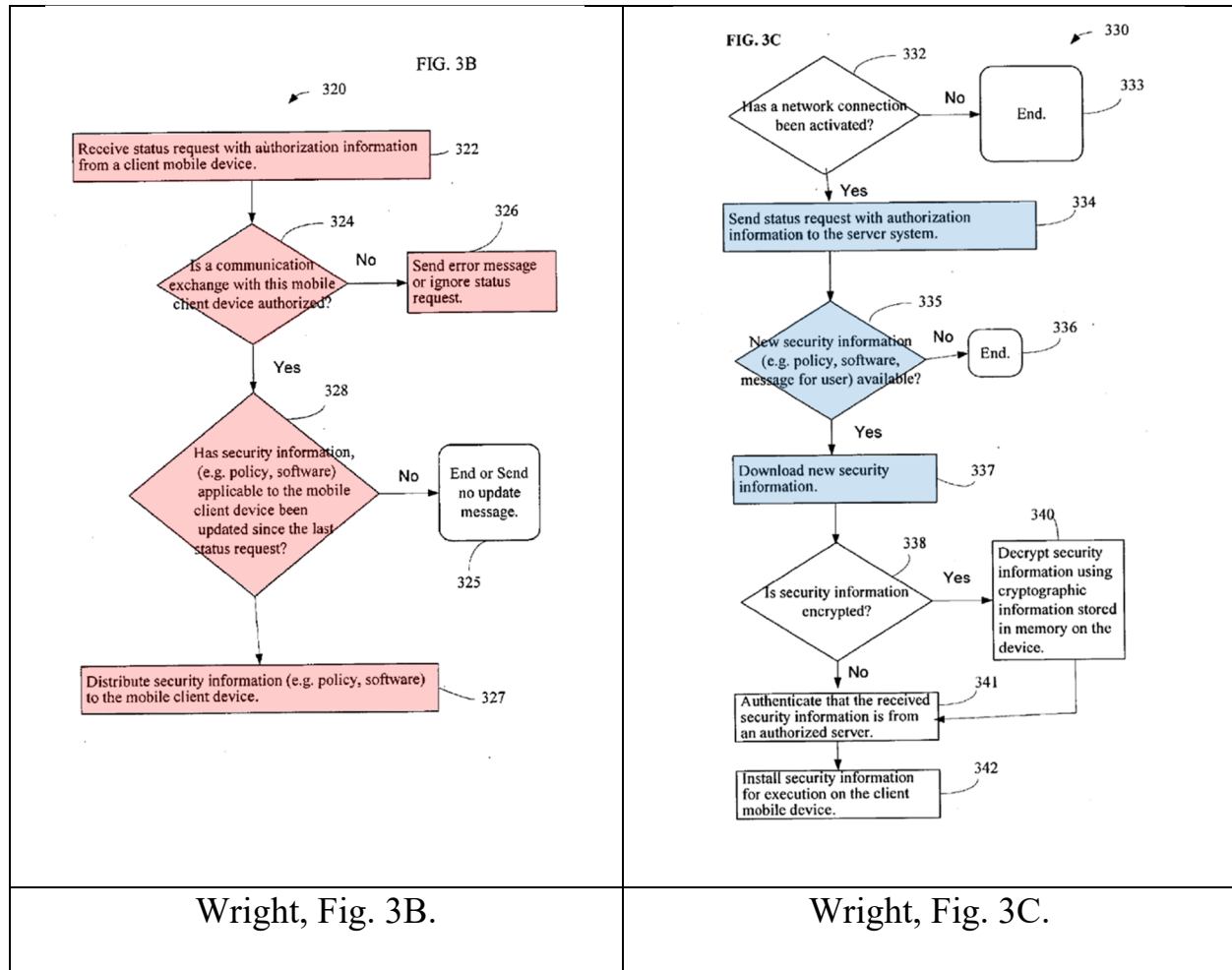
135. Moreover, claim 1 requires only that the device agents *at least assist[] to implement the service policy*. In my opinion, modules 212, 214, and 245 satisfy 1[c][ii] through their assistance in implementing the first portion alone: module 212 receives and installs the security rules, module 245 authenticates them before

installation, and module 214 enforces them. Wright, [0061], [0064], [0068], [0131]–[0132].

f. 1[d]: “using the service processor authentication information, facilitating execution of an end-user device service processor authentication procedure.”

136. In my opinion, Wright discloses or suggests 1[d]. When client 201 sends the SHA1 hash, as I explained for 1[c][i], system 200 uses it to determine whether client 201 (*service processor*) is authenticated. Wright, [0133]–[0137]. System 200 computes “the same SHA1 hash” from the random challenge it issued and the enterprise key, and if that hash does not match client 201’s response, system 200 “drops the connection with no further ado.” *Id.*, [0130], [0137]. If the values match, “the server has authenticated the client.” *Id.*, [0137].

137. This comparison is how system 200 makes the authorization/authentication decision of Figure 3B (step 324) when it uses Establish Session, as I explained for 1[a]. On failure, system 200 drops the connection (step 326). Wright, [0128], [0137]. On success, the exchange proceeds to steps 328 and 327, and system 200 sends any updated policy: policy distribution module 234 determines whether updated security information applies to the device and distributes it, and, in Wright’s protocols, client 201 queries for and downloads the encrypted policy file. *Id.*, [0128], [0138]–[0141], [0144], [0146]–[0147].



138. The comparison verifies client 201's response against the enterprise's key. Wright, [0130]. Because "[c]lients installed by some other enterprise will have a different key, and clients installed in an unmanaged mode will not have a key," a matching response shows that the enterprise's managed client software, client 201, computed it on the device. *Id.* The challenge is random, so the response cannot be computed in advance, and only software holding the enterprise key can compute the value that matches system 200's hash. *Id.*, [0130], [0135]–[0137].

139. When system 200 has “authenticated the client,” it has authenticated client 201 (*service processor*) on the mobile device (*end-user device*). Wright, [0137]. In Wright, the client of this protocol is client 201, the software executing on the mobile device that sends the protocol messages, and the response that system 200 verifies is the SHA1 hash that client 201 computes there, as I explained for 1[c][i]. *Id.*, [0046], [0133]–[0137]. By performing the comparison and deciding whether client 201 is authenticated, system 200 carries out an authentication procedure for the service processor on the end-user device.

140. Accordingly, Wright discloses or suggests that system 200 uses the client’s SHA1 hash response (*using the service processor authentication information*) in a procedure that compares that response with system 200’s own computed SHA1 hash to determine whether client 201 on the mobile device is authenticated (*facilitating execution of an end-user device service processor authentication procedure*). Wright, [0128], [0130], [0133]–[0137].

2. **Claim 2: “The method recited in claim 1, wherein the network system comprises an authentication, authorization, and accounting (AAA) server, a home location register (HLR), a policy charging and rules function (PCRF), an access network authentication system, an admission system, a log-in system, or a combination of these.”**

141. In my opinion, Wright discloses or suggests claim 2. System 200’s authorization module 232 “authorizes a communication exchange between the client

mobile device and the policy distribution or policy management modules.” Wright, [0052]. As I explained for 1[d], system 200 admits client 201 to a session only if authorization/authentication succeeds, and otherwise drops the connection or ignores the request. *Id.*, [0128], [0137]–[0138]. System 200 thus authenticates client 201 on each device that reaches it over a wireless access network when the device’s network access starts up, and it decides whether to admit client 201 to an established session. *Id.*, [0128]–[0129], [0137]–[0138]. In my opinion, a POSITA would have understood that, by performing those functions, system 200 (*network system*) comprises *an access network authentication system* and *an admission system*.

142. System 200 also comprises *a log-in system*. In Establish Session, the client’s first message presents the user identification (UID), and system 200 establishes the session only after verifying the client’s response to its random challenge. Wright, [0134]–[0137]. In my opinion, a POSITA would have understood an exchange that begins with the requester presenting a user identification, and in which the server establishes a session only after a credential check, to be a log-in. System 200, which conducts that exchange, therefore comprises *a log-in system*.

143. Accordingly, Wright discloses or suggests that system 200 (*the network system*) comprises *an access network authentication system*, *an admission system*, and *a log-in system*. Wright, [0052], [0128], [0134]–[0138].

3. **Claim 3: “The method recited in claim 1, wherein the network system comprises a service controller, a gateway, a router, a gateway general packet radio service (GPRS) support node (GGSN), a serving GPRS support node (SGSN), a proxy, a charging element, a notification trigger element, or a combination of these.”**

144. In my opinion, Wright discloses or suggests claim 3. System 200 (*network system*) comprises *a service controller*, a term that the '777 Patent states “may be used to refer to a service processor authentication and management system.” '777 Patent, 48:32–34. System 200 authenticates client 201: as I explained for 1[d], authorization module 232 determines whether client 201 is authorized and, in Establish Session, authenticates it by verifying its SHA1 response. Wright, [0052], [0128], [0133]–[0137]. System 200 also manages client 201: policy management module 236 “manages the one or more client devices for security purposes” and “maintains 314 client management information for the mobile device and the one or more policies associated with it,” as I explained for 1[b]. *Id.*, [0104]. System 200 is therefore a service processor authentication and management system.

145. Accordingly, Wright discloses or suggests that system 200 (*the network system*) comprises *a service controller*. '777 Patent, 48:32–34; Wright, [0052], [0104], [0133]–[0137].

4. **Claim 4: “The method recited in claim 1, wherein facilitating execution of the end-user device service processor authentication procedure comprises verifying that the service processor is present and configured in an expected manner.”**

146. In my opinion, Wright discloses or suggests claim 4. As I explained for 1[d], system 200 authenticates client 201 (*the service processor*) by comparing client 201’s SHA1 hash response with its own SHA1 hash. Wright, [0133]–[0137]. That procedure uses the enterprise key that system 200 generates at installation and distributes to its managed clients for “authentication of clients”; clients installed by another enterprise have a different key, and unmanaged clients “will not have a key.” *Id.*, [0130].

147. Because system 200 sends a “random challenge” in each session, a matching response cannot be replayed. Wright, [0133], [0135]. Wright describes Establish Session as a protocol that “may be used to authenticate the client as a first step in all client-server transactions,” and when it is used, each session begins with a fresh random challenge, so a response recorded from an earlier session does not match the hash of a new challenge. *Id.* A matching response therefore shows that client 201 is running on the device and answering the challenge in the current session (*verifying that the service processor is present*). *Id.*, [0135]–[0137].

148. And because only clients provisioned by the enterprise have the enterprise’s key, a match also shows that client 201 was installed and provisioned as

a managed client (*configured in an expected manner*). When system 200 is installed, a key is generated and stored in an encrypted file that “is copied and distributed to the clients”; “[c]lients installed by some other enterprise will have a different key, and clients installed in an unmanaged mode will not have a key.” Wright, [0130]. In Wright’s design, the enterprise thus provides the key only to the clients it installs and provisions as managed clients, and a client without that key, or with another enterprise’s key, would not produce the expected response. *Id.*, [0130], [0136]–[0137]. In my opinion, a POSITA would therefore have understood that a successful response verifies that client 201 has the configuration with which Wright provisions its managed clients, consistent with Wright’s stated objective of protecting against “unauthorized or rogue mobile devices.” *Id.*, [0052], [0130]. What the match verifies is client 201’s configuration as one of the enterprise’s managed clients, holding the enterprise key, as opposed to a client “installed by some other enterprise” or one “installed in an unmanaged mode.” *Id.*, [0130]. It does not verify whether the device on which client 201 answers the challenge is the device to which system 200 issued the token. As I explained for 1[a], every managed client computes the same response to the same challenge, and a copy of a provisioned client 201 carries the same configuration, including the key file; that is why, as I explain for 8[d] in Ground 1C, Wright-Durham adds Durham’s device-bound check of whether the authenticating device is the device to which system 200 issued the token.

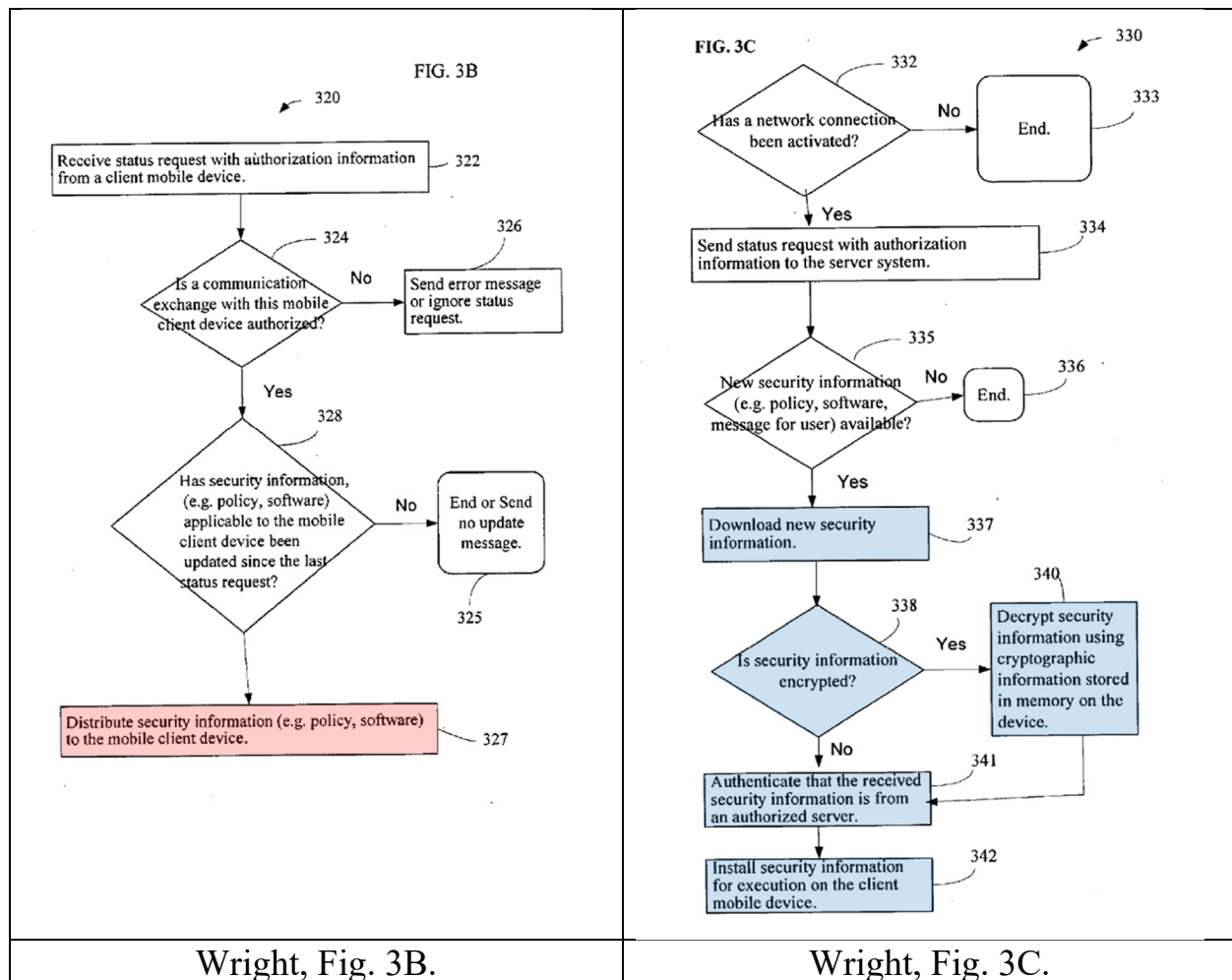
149. Accordingly, Wright discloses or suggests that system 200's challenge-response comparison (*facilitating execution of the end-user device service processor authentication procedure*) verifies that client 201 (*the service processor*) is running on the device and answering the challenge (*is present*) and was installed and provisioned as one of the enterprise's managed clients, holding the enterprise key (*configured in an expected manner*). Wright, [0052], [0130], [0133]–[0137].

5. Claim 5: “The method recited in claim 1, further comprising executing a first portion of the service policy when the end-user device service processor authentication procedure is successful.”

150. In my opinion, Wright discloses or suggests claim 5. As I explained for 1[b], the first portion of Wright's *service policy* is the device's security rules, which system 200 sends to an authorized and authenticated client and which client 201 installs and enforces. Wright, [0048], [0128], [0132], [0145]–[0147].

151. As I explained for 1[d], when authorization/authentication succeeds (*when the end-user device service processor authentication procedure is successful*), system 200 sends any updated security rules, which is executing the first portion (*executing a first portion of the service policy*). Wright, [0128], [0138]–[0141], [0146]–[0147]. In Figure 3B, policy distribution module 234 determines whether updated security information applies to the device and distributes it (steps 328 and 327); in Wright's protocols, client 201 sends the “Query Policy Update” message

and, when a newer policy exists, the “Download Policy” verb, and system 200 returns “the encrypted policy file.” *Id.* Client 201 then installs and enforces the rules: authorization module 245 authenticates the policy, policy setting module 212 installs it, and policy enforcement control module 214 applies the “enforcement mechanism modules as specified by the policy.” *Id.*, [0068], [0132].

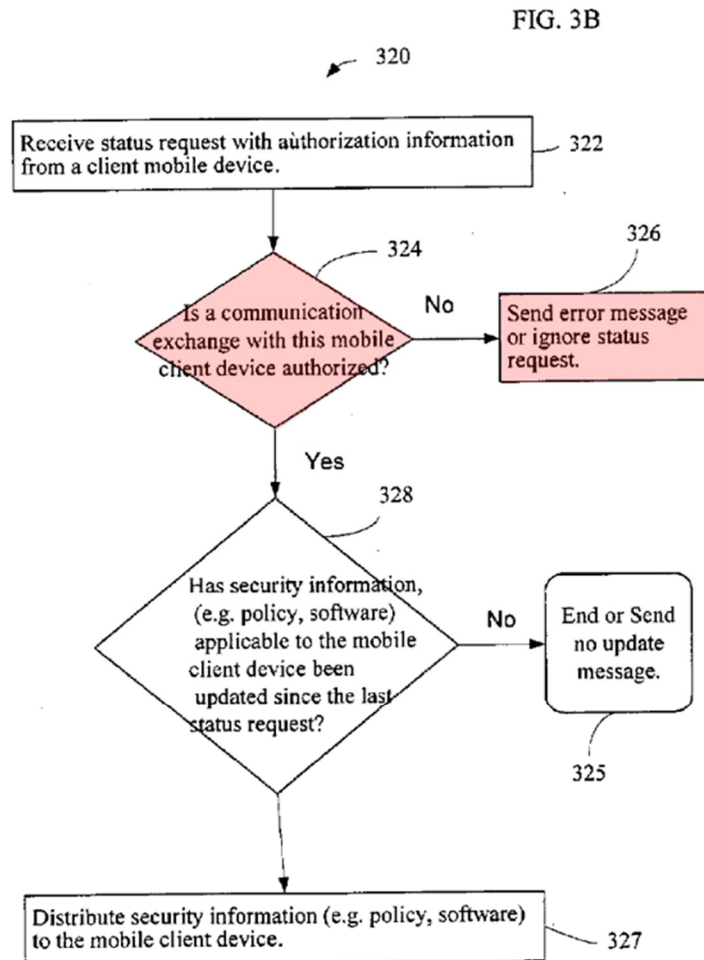


152. Accordingly, Wright discloses or suggests that, when authorization/authentication succeeds (*when the end-user device service processor*

authentication procedure is successful), system 200 sends any updated security rules to client 201, which installs and enforces them (*executing a first portion of the service policy*). Wright, [0068], [0128], [0132], [0138]–[0141], [0146]–[0147].

6. Claim 6: “The method recited in claim 5, further comprising executing a second portion of the service policy when the end-user device service processor authentication procedure is unsuccessful.”

153. In my opinion, Wright discloses or suggests claim 6. When step 324 of Figure 3B fails to authorize/authenticate client 201, system 200 denies the request at step 326: “the status request is ignored 326 or an error message is sent 326.” Wright, [0128]. As I explained for 1[b], that denial is the second portion of the device’s *service policy*.



Wright, Fig. 3B.

154. As I explained for 1[a] and 1[d], when system 200 performs that step with Establish Session, failure means that client 201's SHA1 response does not match system 200's SHA1 hash (*when the end-user device service processor authentication procedure is unsuccessful*). Wright, [0137]. System 200 then "drops the connection with no further ado." *Id.*; *see id.*, [0128]. That denies client 201 access

to system 200, including the session through which it would otherwise query for and download policy information, which is executing the second portion (*executing a second portion of the service policy*). *Id.*, [0137]–[0147].

155. Accordingly, Wright discloses or suggests that, when client 201's SHA1 response does not match (*when the end-user device service processor authentication procedure is unsuccessful*), system 200 drops the connection and denies client 201 access to system 200 (*executing a second portion of the service policy*). Wright, [0128], [0137].

7. **Claim 7: “The method recited in claim 1, wherein the device credential comprises information from a subscriber identity module (SIM) card, information from a soft SIM, information from a universal SIM, an international mobile subscriber identity (IMSI), a wireless modem identifier, a phone number, an international mobile equipment identity (IMEI), a mobile equipment identifier (MEID), a media access control (MAC) address, an Internet protocol (IP) address, a device identifier, an encryption key, or a combination of these.”**

156. In my opinion, Wright discloses or suggests claim 7. A POSITA would have understood that Wright's encrypted token is a per-device credential that identifies the device to system 200 (*device credential comprises ... a device identifier*), as I explained for 1[a]. Wright, [0078]. Authorization module 232 provides the token “to the mobile client device during a trusted connection between the two,” so each token is issued to a particular device, and, as I explained for 1[a],

a POSITA would have understood that system 200 receives the token back as the authorization information accompanying the status request. *Id.*, [0078], [0128]–[0129].

157. Because system 200 issued the token, it would know which device has the token and would recognize the token as a *device identifier*. System 200 provisioned the token to the device in a one-to-one exchange, “during a trusted connection between the two,” and maintains client management information for that device. Wright, [0078], [0104]. In Wright’s design, system 200 provisions each token to one device, so, as Wright provisions the token, the device that has the token is the device to which system 200 issued it: client 201 on that device holds the token and presents it as the authorization information with its status requests. *Id.*, [0078], [0128]–[0129]. The token need not contain a readable hardware address or other embedded identifier; the token itself is the value that system 200 issued to, and associates with, the provisioned device, so its presentation identifies to system 200 the device to which it was issued. *Id.*, [0078], [0104]. A device running a copy of client 201 would also hold and present that token even though system 200 did not issue the token to it, as I explain for 8[d] in Ground 1C; the token would remain the identifier of the device to which system 200 issued it, which is why Wright-Durham adds Durham’s device-bound check of whether the device presenting the token is that device.

158. Accordingly, Wright discloses or suggests that system 200 receives an encrypted token that identifies the device to system 200 (*device credential comprises ... a device identifier*). Wright, [0078], [0104], [0128]–[0129].

8. Claim 10: “The method recited in claim 1, wherein the service processor comprises a client configured to implement at least a portion of the service policy.”

159. In my opinion, Wright discloses or suggests claim 10. Client 201 (*service processor*) is a *client* that “enforces the policies set up by the administrator.” Wright, [0046]. It implements the first portion of the *service policy*. Its policy enforcement module 214 “comprises instructions for enforcing the security policy currently set by the policy setting module 212” and does so through “enforcement mechanism modules as specified by the policy.” *Id.*, [0064], [0068]. Before enforcement, policy setting module 212 receives the policy, authorization module 245 authenticates it, and module 212 installs it “for execution on the client mobile device,” as I explained for 1[c][ii]. *Id.*, [0061], [0132]. Thus, client 201 is a *client configured to implement at least a portion of the service policy*.

160. Accordingly, Wright discloses or suggests that client 201 (*the service processor*) is a client that enforces the policies set up by the administrator (*comprises a client*), and that its modules 212, 214, and 245 receive, authenticate, install, and enforce the security rules (*configured to implement at least a portion of the service policy*). Wright, [0046], [0058], [0061], [0064], [0068], [0132].

9. Claim 11: “The method recited in claim 1, wherein the service policy comprises a control policy, a service usage limit, an accounting policy, a billing policy, a notification policy, or a combination of these.”

161. In my opinion, Wright discloses or suggests claim 11. The first portion of Wright’s *service policy*, the security rules that module 214 enforces, “may not allow a particular network service or application or both to be used” (*comprises a control policy*). Wright, [0014]; *see id.*, [0048]. Those rules control which network services and applications the device may use. The second portion also comprises *a control policy*: it denies client 201 access to system 200 when authentication fails. *Id.*, [0128], [0137]. Thus, the *service policy* as a whole controls the device’s access to network services, applications, and system 200 (*the service policy comprises a control policy*), as I explained for 1[b].

162. Accordingly, Wright discloses or suggests that its *service policy*, comprising security rules that control the device’s use of network services and applications and a denial that controls the device’s access to system 200, comprises *a control policy*. Wright, [0014], [0048], [0128], [0137].

10. Claim 13: “The method recited in claim 5, wherein executing the first portion of the service policy comprises sending the first portion of the service policy to the service processor.”

163. In my opinion, Wright discloses or suggests claim 13. As I explained for claim 5, system 200 executes the *first portion* by sending the updated security rules to client 201 (*service processor*). Wright, [0128], [0138]–[0147]. After

Establish Session, client 201 sends the “Download Policy” verb, and system 200 responds with “the encrypted policy file,” which contains the device’s security rules. *Id.*, [0145]–[0147]. Policy setting module 212 of client 201 receives and installs that policy. *Id.*, [0061], [0131]–[0132]. Thus, system 200’s execution of the first portion consists of sending it to client 201 (*executing the first portion of the service policy comprises sending the first portion of the service policy to the service processor*).

164. Accordingly, Wright discloses or suggests that system 200 executes the first portion by sending the updated security rules to client 201 (*executing the first portion of the service policy comprises sending the first portion of the service policy to the service processor*). Wright, [0128], [0138]–[0147].

11. **Claim 14: “The method recited in claim 6, wherein the second portion of the service policy differs from the first portion of the service policy in a network access privilege, an allowable network destination, a service usage accounting or billing policy, a service usage rate, a service speed or quality, an allowed access network, a user notification, a roaming policy or permission, an effect on a device operation, or a combination of these.”**

165. In my opinion, Wright discloses or suggests claim 14. The two portions described for 1[b] differ in *a network access privilege*. Under the *first portion*, client 201 keeps its connection to system 200 and downloads policies through it, subject to security rules that “may not allow a particular network service or application or both to be used.” Wright, [0014], [0138], [0146]–[0147]. Under the *second portion*,

system 200 “drops the connection with no further ado” and denies access to system 200 altogether. *Id.*, [0137]; *see also id.*, [0128]. The device’s privilege to access system 200, and through it to receive policies, therefore differs between the two portions.

166. For the same reason, the portions differ in *an effect on a device operation*. The *first portion* restricts which services and applications the device may use, through the enforcement mechanisms that module 214 applies, while the *second portion* ends the device’s connection to system 200. Wright, [0014], [0068], [0128], [0137].

167. Accordingly, Wright discloses or suggests that the denial of access (*the second portion of the service policy*) differs from the security rules (*differs from the first portion of the service policy*) in the device’s access to system 200 (*in a network access privilege*) and in their effect on the device’s operation, because the security rules restrict the services the device may use while the denial ends its connection (*an effect on a device operation*). Wright, [0014], [0068], [0128], [0137]–[0147].

B. GROUND 1B: CLAIMS 15–17, 19–22, 28, 29, 31, AND 32 ARE OBVIOUS OVER WRIGHT AND BITTMANN

1. Bittmann’s Teachings

168. As explained in the Overview of Bittmann, Bittmann describes a client-server system that monitors a device’s network activity, reports it to a server, and uses the reports to account for usage and restrict further activity. Bittmann, Abstract,

[0042]–[0045]. Three aspects of those teachings matter for this ground. First, on the device, activity monitoring unit 130, “installed within communication device 110 or ... connected thereto,” “monitors data activity of communication device 110,” and its report generator 420 sends server utility 150 reports that can identify the volume of data transferred, the time interval of the transfer, its source and destination, the application using the data, and the application protocol. *Id.*, [0028], [0054], [0059]–[0060], [0062]. Second, within server utility 150, mediation system 210 receives those reports and other reports of activity and correlates reports from multiple sources into a unified report record; rating system 220 rates the reported activity, including “data activity reported by system resources (for example probes, switches, VoIP servers, etc.)”; and allowance manager 230 updates the allowances of the account associated with the device based on that rating. *Id.*, [0042]–[0043]. Third, an allowance counts activity, such as “volume in bytes,” “number of accesses,” or “number of downloads,” and when an allowance reaches its “pre-determined restriction level,” server utility 150 restricts further activity, which control 430 in each monitoring unit applies. *Id.*, [0038], [0045], [0055], [0070]–[0071].

2. Motivation to Combine Wright and Bittmann

169. In my opinion, a POSITA would have been motivated to incorporate Bittmann’s usage monitoring, reporting, rating, and account management into Wright’s system 200 and client 201. I explain below what Wright describes and what

it does not detail, the two reasons a POSITA would have combined the references, how the combined system would operate, and why a POSITA would have had a reasonable expectation of success.

170. Wright authenticates client 201 when a matching SHA1 response shows that client 201 held the enterprise's key when the session began. When system 200 is installed, a key is generated, stored in an encrypted file, and "copied and distributed to the clients," and "[c]lients installed by some other enterprise will have a different key." Wright, [0130]. In the Establish Session protocol, system 200 sends client 201 a "random challenge," client 201 returns "SHA1(random challenge, key)," and system 200 "computes the same SHA1 hash" and compares the two; if they match, "the server has authenticated the client." *Id.*, [0134]–[0137]. In my opinion, that check does not show that client 201 was not compromised afterward, including in how it reports its activity. Establish Session authenticates the client "as a first step in all client-server transactions": it verifies client 201 when client 201 answers the challenge at the start of the session, and it does not examine the messages, including activity reports, that client 201 sends afterward in that session. *Id.*, [0133], [0145]. If client 201's software were tampered with after the session began, the authentication already completed would not reveal it, and nothing in the protocol tests whether the activity client 201 reports is accurate.

171. Wright's policies specify what client 201 monitors, but Wright does not detail how usage information would be used to track or limit a device's use of a service. Policy management module 236 defines policies that specify, among other things, "ports to be monitored, network services to be monitored, applications to be monitored, enforcement mechanisms to be put in place for a particular policy." Wright, [0048]. Wright's server "allows an administrator to ... analyze logs," and in Wright's Figure 8 embodiment the client "provides its status, usage and error logs to the server." *Id.*, [0046], [0220]. Wright's policies determine whether a device may use a service at all: a security policy "may not allow a particular network service or application or both to be used" based on the device's location or the activity status of a security feature. *Id.*, [0014]. But Wright does not detail what its usage logs contain or how system 200 uses them, for example to track how much of a permitted service a device uses or to limit that use. *Id.*, [0014], [0220].

172. Bittmann describes those details. Bittmann's report generator 420 sends server utility 150 reports that can identify, among other things, the "volume of data transferred (e.g. kilobytes, number of packets, etc)," the "time-interval for data transfer," the "source and destination of data," "the application on communication device 110/140 using the data," and "the application protocol." Bittmann, [0054], [0060]. Server utility 150 uses those reports to track and limit the device's usage: mediation system 210 receives and correlates the reports, rating system 220 rates the

reported activity, allowance manager 230 updates the allowances of the account associated with the device, and when an allowance reaches its “pre-determined restriction level,” allowance manager 230 “causes the restriction of activity in the account.” *Id.*, [0042]–[0045]. Bittmann also explains that its reports are useful “for rating purposes and/or for fraud and network management,” and it seeks to protect them from tampering “so as to ensure integrity of the activity data transmitted to server utility 150.” *Id.*, [0057], [0059]. Bittmann thus describes what a device’s usage report contains and how a server uses those reports to track and limit the device’s use of network services, which are the details Wright does not provide.

173. In my opinion, a POSITA would have been motivated to combine Wright and Bittmann for at least two reasons, which I explain in turn. First, Bittmann’s reporting and tracked allowances would let Wright’s administrator monitor and limit how much each device uses a permitted service. Second, comparing a device’s own reports with independent reports of the same activity would give system 200 another way to detect a client compromised after authentication.

174. First, Bittmann’s reporting, either from the device alone or from both the device and a trusted source, together with Bittmann’s tracked allowances, would let Wright’s administrator monitor and limit how much each device uses a permitted service, for example to manage network capacity and data consumption. Bittmann’s

allowances measure use: an allowance can represent “a quantity of past or future activity, for example volume in bytes, volume in packets, time units, number of accesses, number of downloads,” and when an allowance reaches its “pre-determined restriction level,” allowance manager 230 causes the restriction of activity in the account; if an allowance represents “the number of transmitted emails,” for example, only emails may be restricted. Bittmann, [0038], [0045]. Allowance manager 230 can track an allowance from the device’s reports alone or from those reports together with reports from other sources. At a minimum, it receives the rating of the activity that monitoring unit 130 reports, and it may also receive the rating of “data activity reported by system resources (for example probes, switches, VoIP servers, etc).” *Id.*, [0043]. Wright’s policies determine whether a device may use a service, and Wright’s client provides “usage ... logs,” but Wright does not detail measuring or capping how much of a permitted service a device consumes. Wright, [0014], [0048], [0220]. With Bittmann’s allowances, the administrator would be able to cap, for example, the volume of data each device transfers over the enterprise’s network or the number of emails it sends, so that no device consumes a disproportionate share of the network’s capacity. In the combination, the administrator would set each device’s allowances and restriction levels as part of the policy that system 200 associates with the device in its client management record, so that the device’s policy would govern not only whether the

device may use a service but also how much of it the device may use before activity is restricted. Wright, [0048], [0104]; Bittmann, [0038], [0045].

175. Second, providing device reports and independent reports would give system 200 another way to detect a client compromised after authentication. Wright is concerned with compromised clients. Authorization module 232 is “a further safeguard against unauthorized or rogue mobile devices trying to hijack the security policies or corporate data,” and Wright’s protective components report whether other services “have been disabled or removed from the system,” information that “may be relayed to the policy server 850 as it indicates a possible compromise of the protective system.” Wright, [0052], [0255]. In my opinion, a client tampered with after authentication would be able to misreport its activity. Client 201 generates its own reports, so a tampered client 201 can understate or omit the data it transfers, for example while it retrieves large volumes of corporate data, and its reports would then show nothing unusual. Bittmann recognizes that risk for device-generated reports: it describes “safety features intended to block the ability to tamper with the device or with the automatic transmission feature, so as to ensure integrity of the activity data transmitted to server utility 150.” Bittmann, [0057]. Those safety features seek to prevent tampering with the device and its reporting; they do not give the server a measurement, independent of the device, against which to check the device’s reports.

176. A probe or server that measures the same activity would provide reports that the client cannot alter. Bittmann's mediation system 210 receives "other reports of activity" and correlates "reports from multiple sources," and Bittmann identifies probes, switches, and VoIP servers as system resources that report data activity. Bittmann, [0042]–[0043]. A probe on the enterprise's network measures the device's traffic as that traffic crosses the network, and a server measures the traffic it exchanges with the device as it provides its service. Neither the probe nor the server relies on client 201 to measure or report the traffic, so altering client 201 or its reports would not change their measurements. Comparing the client's report with the independent report would reveal misreporting, and therefore compromise, that the authentication at the start of the session does not detect: a client that understates or omits activity would report less than the independent source measured for the same traffic. To make that comparison, the independent source would record the same quantities that the device reports for its activity with the network, such as the "volume of data transferred," the "source and destination of data," "the application protocol," and the "time-interval for data transfer." *Id.*, [0060]. Recording the same quantities would let mediation system 210 line up the two reports for the same traffic and compare like with like.

177. Bittmann identifies the independent sources: "system resources (for example probes, switches, VoIP servers, etc)." Bittmann, [0043]. Wright's

enterprise network includes servers that handle a managed device's traffic and would serve as such sources. Wright describes a "corporate mail server" to which email between users on the same corporate domain is directed, and its location-detection example identifies the enterprise's POP3 mail service by the server's address and port. Wright, [0161], [0201]–[0202]. Wright's policy server may also cache files transferred between clients. *Id.*, [0161]. And Wright's users retrieve corporate data over that network: "an executive could be retrieving sensitive files or emails from the corporate network." *Id.*, [0009]. In my opinion, a POSITA also would have understood that a corporate Intranet such as Wright's network would commonly provide a VoIP server. Cho, published May 18, 2006, explains that "[a]t present, a representative example of the VoIP technique includes the IP telephony service which is mainly provided over an Intranet network using an IP-PBX." Cho, [0006]. Cho's statement shows that, by 2006, the representative form of VoIP was IP telephony that an organization ran over its own Intranet through an IP-PBX, the server that sets up and manages the organization's IP telephone calls. A VoIP server on a corporate Intranet was therefore a conventional configuration, not an unusual one, and an enterprise network like Wright's that carried its users' voice traffic would have done so through such a VoIP server. Bittmann likewise lists "voice over IP" among the data activities that monitoring unit 130 monitors and "VoIP servers" among the system resources that report data activity. Bittmann, [0031], [0043]. Each

of these servers handles the device's traffic for the service it provides and therefore can measure and report that traffic.

178. Bittmann also contemplates that these resources may be the enterprise's own or provided by others. Bittmann's communication infrastructure, which includes "all components of communication network 120 that are involved in the provision of communication service to communication device 110," "can be thought of as 'belonging' to one or more network operators (either by ownership, rental, or other suitable arrangement)." Bittmann, [0034]–[0035]. Bittmann's examples of that infrastructure include not only routers but also "name servers" and "end nodes," and Bittmann identifies probes, switches, and VoIP servers as "system resources" that report data activity. *Id.*, [0034], [0043]. In my opinion, a POSITA would have understood a probe on the enterprise's network and the enterprise's mail and VoIP servers to be components of that kind: each is part of network 120, and each is involved in providing the device's service, the servers by providing the device's email and telephony services and the probe by metering the device's traffic, a function Bittmann associates with "network management." *Id.*, [0059]. Network 120 may be an "Intranet," and server utility 150 may be "comprised within or integrated within the system of the network operator, for example in the internal network of the network operator." *Id.*, [0032], [0041]. In my opinion, a POSITA would have understood that when Bittmann's network is an Intranet, the enterprise that runs it is

the network operator. Bittmann thus contemplates that the probe and servers may be owned by the enterprise, rented from a third party, or provided under another arrangement.

179. In the combination, client 201 would include monitoring unit 130, which would report the mobile device's activity with Wright's network during a session to system 200. Bittmann's monitoring unit 130 may be installed within the communication device, "monitors data activity of communication device 110," and, through report generator 420, reports that activity to server utility 150 "automatically in a manner not controlled by the user." Bittmann, [0028], [0054], [0062], Fig. 1. Monitoring unit 130 may be implemented in software and placed where the device's traffic passes between the CPU and the network interface. *Id.*, [0052]–[0053]. In client 201, layer manager 206 "processes network traffic," so monitor 410 would observe the traffic that layer manager 206 processes. Wright, [0060]. Bittmann permits monitoring unit 130 to monitor all or only some of the device's data activity, and the monitored activity includes, among other things, "emails" and "voice over IP." Bittmann, [0031]. In the combination, monitoring unit 130 would monitor and report the device's activity with Wright's network, that is, the traffic the device exchanges with the enterprise's network and its servers during a session.

180. Client 201's report would arrive in the session the device opens with its encrypted token, which identifies the device's client management record. As

explained for 1[a] and 1[b], client 201 opens each session by sending its token, and the token identifies the client management information that policy management module 236 “maintains” for the device “and the one or more policies associated with it.” Wright, [0078], [0104]. Establish Session authenticates the client “as a first step in all client-server transactions,” so every report client 201 sends within the session would be associated with the token that opened it. *Id.*, [0133]. The probe and server reports would cover the traffic of that same session, so system 200 would associate them with the same token and record. Client 201 sends that token, as authorization information in its status request, when the device’s network access starts up (Wright, [0128]–[0129]), and the device then sends and receives its traffic from the network address with which it opened the session. A probe or server identifies the traffic it measures by that address and by the time interval of the measurement, and the device’s own report carries the same time interval. Bittmann, [0060]. System 200 would therefore match each probe or server report to the token-opened session from that address during that interval, which is the session in which client 201’s report for the same interval arrives, and, through that session’s token, to the device’s client management record. Wright, [0078], [0104], [0133].

181. System 200 would include server utility 150, with mediation system 210, rating system 220, and allowance manager 230. Bittmann, [0042], Fig. 2. Bittmann permits server utility 150 to be “any combination of hardware, software

and/or firmware,” to be located inside or outside the network cloud, and to be “comprised within or integrated within the system of the network operator, for example in the internal network of the network operator.” *Id.*, [0030], [0041]. These components would therefore run as additional modules of system 200, which “may be implemented by a series of networked computers as may typically be implemented by an enterprise.” Wright, [0046]–[0047]. Within system 200, mediation system 210 would receive the reports from monitoring unit 130 and from the probe or servers and reconcile them, rating system 220 would rate the reported activity using the rates and plan information stored in database 240, allowance manager 230 would update the allowances of the account associated with the device, and system 200 would take the other steps I describe for the individual claims below. Bittmann, [0042]–[0043]. When an allowance reaches its restriction level, system 200 would instruct monitoring unit 130 to restrict the activity the allowance governs, for example the device’s emails when the allowance counts “the number of transmitted emails,” and control 430 would apply the restriction, as Bittmann’s server utility 150 does. *Id.*, [0045], [0055], [0070]–[0071].

182. Because the reporting would take place within the sessions that Establish Session authorizes and authenticates, claim 1’s steps remain as I described them in Ground 1A. System 200 would still receive the device’s encrypted token, obtain the policy applicable to the device, compare client 201’s SHA1 response with

the hash that system 200 computes itself, and either send the security rules of the policy or drop the connection, as explained for claim 1. Wright, [0078], [0104], [0128]–[0129], [0133]–[0147]. Bittmann’s reports would be additional messages that client 201 sends after a session is established, and they do not change how system 200 identifies, authorizes, or authenticates client 201.

183. In my opinion, the combination would have used known techniques (Bittmann’s data monitoring, reporting, and usage allowances) to improve a similar system (Wright’s system 200 and the mobile device with client 201, a client-server system that manages mobile devices’ use of network services) in the same way. Each of Bittmann’s techniques would perform the same function in the combination that it performs in Bittmann: monitoring unit 130 would measure and report the device’s activity, and mediation system 210, rating system 220, and allowance manager 230 would correlate the reports, rate the activity, and update the account’s allowances. Bittmann, [0042]–[0045], [0054]. The result would have been predictable: system 200 would measure and verify the device’s usage and adjust its allowances.

184. In my opinion, a POSITA would have had a reasonable expectation of success. Wright’s protocols “are designed to be extensible and straightforward to allow addition of new verbs later,” so client 201 and system 200 would carry usage reports as additional messages within an established session. Wright, [0133]. Bittmann “is not restricted to any specific system architecture for server utility 150,”

and it permits monitoring unit 130 and server utility 150 to be implemented in “any combination of hardware, software and/or firmware.” Bittmann, [0041], [0042], [0053]. Wright’s client and server already exchange management information of a similar kind: monitoring module 226 “receives diagnostic information such as events or audit logs from a client device,” and the client in Wright’s Figure 8 embodiment provides “usage and error logs” to the server. Wright, [0054], [0220]. Bittmann identifies the information its monitoring unit reports and permits “standard handshake protocols” for the exchange. Bittmann, [0060], [0062].

185. Implementing the combination would have required adding Bittmann’s monitoring and control functions to client 201; adding Bittmann’s mediation, rating, and allowance-management functions to system 200; extending the client-server protocols to carry the reports and restriction instructions; having the enterprise’s probe or servers report the device’s activity to mediation system 210; and associating each report with the session’s token and the device’s client management record. Each step uses functions the references already describe. Wright’s client already sends diagnostic information to the server and installs server-distributed policies, and Wright’s server already maintains a client management record for each device. Wright, [0054], [0059], [0061], [0104]. Bittmann’s monitoring unit already sends activity reports to its server and receives restriction instructions from it, and

Bittmann's system resources already report data activity to the server. Bittmann, [0042]–[0043], [0054], [0062], [0070]–[0071].

3. Claim 15

- a. **15[a]: “The method recited in claim 1, further comprising: receiving a data usage report associated with the end-user device from a good customer feedback source; and”**

186. In my opinion, Wright-Bittmann teaches 15[a].

187. The '777 Patent does not define a *good customer feedback source*, and in my opinion a POSITA would not have read the term as limited to feedback from a customer or to feedback from a third-party business or website. The patent uses “good customer feedback” for the report that a service-providing server or website sends about the device's interactions with it, such as visits, transactions, business generated, or data communicated, not for an opinion that a customer expresses. '777 Patent, 65:17–32, 67:58–68:13. The claims are consistent with that reading: claim 20 recites that “the good customer feedback source comprises a website or server that tracks,” among other things, “an amount of data communicated by or to the end-user device.” Nothing in the patent's description requires that the server or website belong to a business other than the operator of the network that uses its report.

188. The '777 Patent describes, “[i]n some embodiments,” a “trusted source used to validate micro-CDR service usage classification reports” that “is a server or website that provides the service,” where “the validation is provided in the form of

good customer feedback associated with a user credential, a service processor credential or a device credential that can be used to determine which device or user to provide the credit to.” ’777 Patent, 65:17–23. In the patent’s example, “the website servers track the number of visits, number of transactions, amount of business generated, amount of data communicated or another measure of device interaction with the website,” and “a summary of this device interaction with the website can be communicated to the service controller.” *Id.*, 65:17–32. The ’777 Patent also describes using good customer feedback “directly without a service processor on the device”: the website or server “creates a summary report of this device interaction with the website or server” and communicates it to a service controller, which “can then reconcile the good customer feedback summary report.” *Id.*, 67:58–68:13. In my opinion, each of these examples describes a server or website that provides a service and reports the device’s use of that service in a report tied to a credential, which the network relies on because the server or website observed the device’s use itself, independently of the device.

189. Although the ’777 Patent’s non-limiting description at 65:17–23 uses the source’s report to credit a device or user, the claims do not limit the source’s report to crediting. Claim 15 recites using the report to adjust a user account: “based on the data usage report, adjusting a user account.” In my opinion, a POSITA would have understood that adjusting an account encompasses either increasing or

decreasing it. Even the '777 Patent's own good-customer-feedback example adjusts the account in both directions: the service controller reconciles the summary report "by applying a user service usage credit rating rule to deduct a bulk portion of service usage from the user account and add a classification of service usage to the user account that is rated for billing purposes by a rating rule for the given micro-CDR classification." '777 Patent, 67:58–68:13. The rule removes usage from the account in one respect and adds usage, rated for billing, in another.

190. Wright's system 200 "may be implemented by a series of networked computers as may typically be implemented by an enterprise." Wright, [0046]. Wright describes a client providing "usage ... logs" to the server, but, as explained in the motivation-to-combine section of this ground, Wright does not detail what those logs contain or how system 200 uses them, and it does not detail receiving usage reports from its servers. *Id.*, [0220]. Bittmann, however, teaches that server utility 150 receives "data activity reported by system resources (for example probes, switches, VoIP servers, etc)," in addition to the device's own reports. Bittmann, [0042]–[0043]. In the combination, system 200 would include server utility 150, and the enterprise's server, for example Wright's corporate mail server or a VoIP server deployed on the enterprise's network, would report the mobile device's data activity to mediation system 210 in server utility 150. *Id.*; Wright, [0161], [0201]–[0202]. The corporate mail server, for example, would report the email data it delivered to

and accepted from the device during the device's session. As explained in the motivation-to-combine section of this ground, a POSITA would have been motivated to obtain these reports both to track and limit the device's usage and to check the device's own reports.

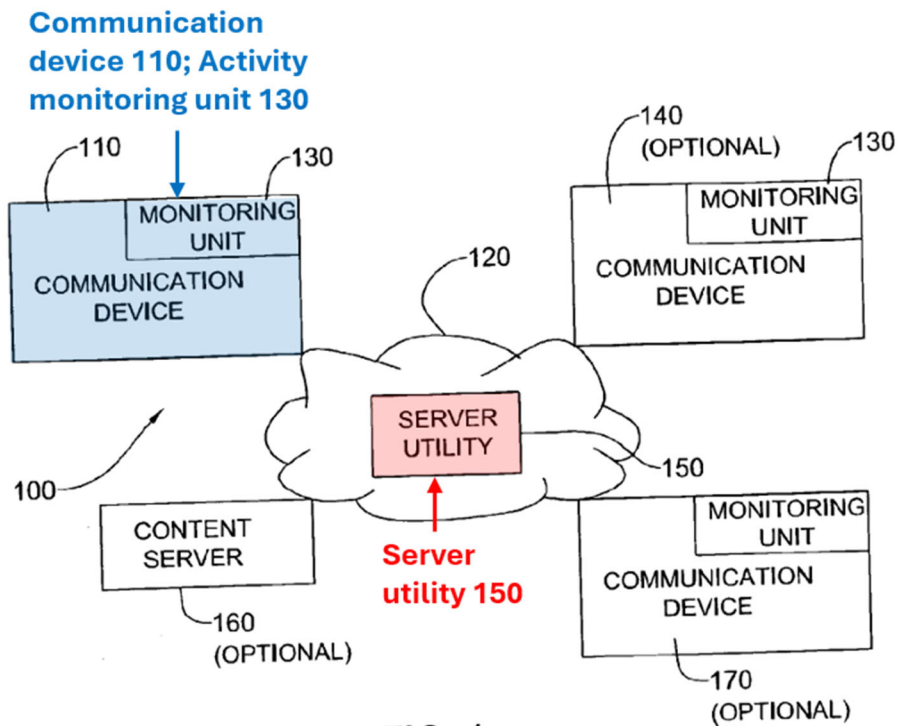


FIG. 1

Bittmann, Fig. 1.

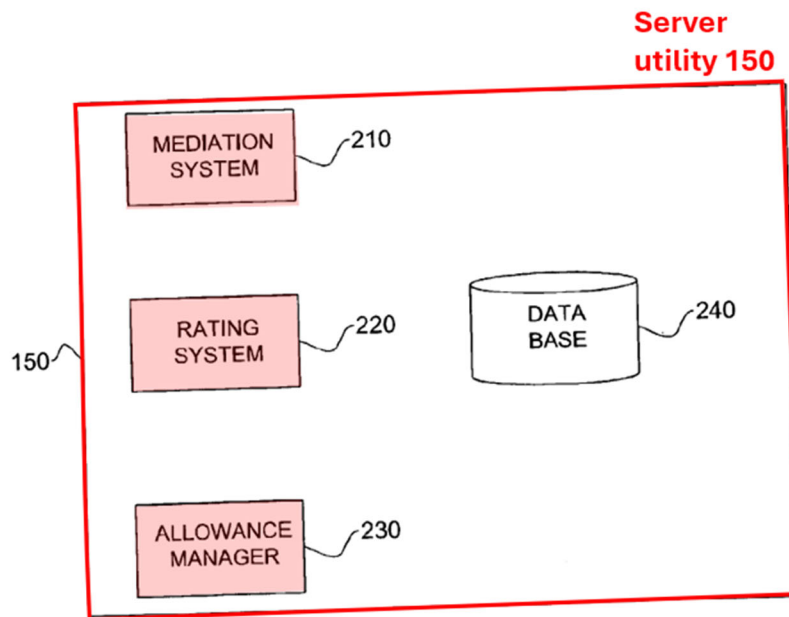


FIG. 2

Bittmann, Fig. 2.

191. As explained for 1[a], client 201 opens each session with its token, and Establish Session authenticates the client “as a first step in all client-server transactions.” Wright, [0128], [0133]. A server reporting the device’s activity would therefore report it for that session, and system 200 would associate the report with the session’s token (the *device credential*) and with the device’s client management record, which, as explained for 1[b], the token identifies. *Id.*, [0078], [0104]. The motivation-to-combine section of this ground explains how system 200 would match the report to the session.

192. A server’s report covers the device’s “data activity,” and mediation system 210 correlates server and device reports “to produce a unified report record” for rating. Bittmann, [0042]–[0043]. In my opinion, a POSITA would have

understood that, so that mediation system 210 would be able to correlate the two reports, the server's report would include the same usage quantities as the device's report, such as the "volume of data transferred." *Id.*, [0060]. Correlating two reports of the same activity requires comparable fields: mediation system 210 can compare or combine a server's count with the device's count only if both count the same thing for the same traffic.

193. In my opinion, as the '777 Patent uses the term, reports from these servers are a *data usage report associated with the end-user device from a good customer feedback source*. Each server is "a server or website that provides the service": the corporate mail server provides the enterprise's email service, and a VoIP server provides its IP telephony service. '777 Patent, 65:17–23. And each report would be associated with the device credential. The '777 Patent describes good customer feedback "associated with a user credential, a service processor credential or a device credential that can be used to determine which device or user to provide the credit to." *Id.* In the combination, system 200 would associate each server report with the session the device opened with its encrypted token (the *device credential*) and would use that token to determine which device's account the report concerns. Each server would also report the device's usage from its own handling of the device's traffic, independently of the device.

194. Moreover, to the extent Wright's enterprise servers were rented or provided under a "suitable arrangement," Wright-Bittmann teaches receiving those reports from a server owned by a third party. Bittmann, [0035]. As explained in the motivation-to-combine section of this ground, Bittmann contemplates that the operator's infrastructure may belong to it "by ownership, rental, or other suitable arrangement," and in the combination the enterprise would be the operator. *Id.*, [0032], [0035], [0041]. A server owned by a third party and rented by the enterprise would provide the same service to the device, handle the same traffic, and report the device's data activity to mediation system 210 in the same way.

195. Accordingly, in my opinion, Wright-Bittmann teaches that system 200 receives the reports of servers providing a service regarding a mobile device's data usage (*receiving a data usage report associated with the end-user device from a good customer feedback source*). Bittmann, [0035], [0042]–[0043]; Wright, [0133], [0161], [0201]–[0202]; '777 Patent, 65:17–23.

b. 15[b]: "based on the data usage report, adjusting a user account, the user account being associated with the end-user device."

196. In my opinion, Wright-Bittmann teaches 15[b].

197. As explained for 15[a], system 200 would receive an enterprise server's report of the mobile device's data activity (*data usage report*). Rating system 220 rates "data activity reported by system resources (for example probes, switches,

VoIP servers, etc),” and allowance manager 230 “updates the allowance(s) for the account associated with communication device 110 ... based on the rating of activity involving communication device 110.” Bittmann, [0043]. An allowance can represent, for example, “volume in bytes,” a “number of accesses,” or a “number of downloads.” *Id.*, [0038].

198. In the combination, allowance manager 230 would use the rating of the server’s report to update the account associated with the mobile device to reflect the device’s usage, for example by reducing a volume allowance by the data that the server delivered to the device, so that the administrator can track that usage against the limits described as the first reason in the motivation-to-combine section of this ground. To update the account, system 200 would use the device’s token to associate the report with the device’s client management record, as explained for 15[a]. That record identifies the device’s user account: policy management module 236 maintains client management information for the mobile device “and the one or more policies associated with it,” and Wright’s example list of client management information fields includes a “User Name” field. Wright, [0104]–[0105]. In the combination, the account whose allowances allowance manager 230 maintains in database 240 would be the account identified by that record. Bittmann, [0042]–[0043].

199. Accordingly, in my opinion, Wright-Bittmann teaches that allowance manager 230, using the rating of the enterprise server's report (*based on the data usage report*), updates the allowance of the account (*adjusting a user account*) that system 200 identifies through the mobile device's token and client management record (*the user account being associated with the end-user device*). Bittmann, [0038], [0043]; Wright, [0104]–[0105].

4. Claim 16

a. 16[a]: “The method recited in claim 1, further comprising: receiving a device data usage report from the end-user device”

200. In my opinion, Wright-Bittmann teaches 16[a].

201. In the combination, client 201 would include Bittmann's monitoring unit 130, and system 200 would include Bittmann's server utility 150, as explained in the motivation-to-combine section of this ground. Bittmann permits monitoring unit 130 to be “installed within communication device 110” and to be implemented in software, and client 201 is software executing on the mobile device (*the end-user device*). Bittmann, [0028], [0053]; Wright, [0046]. Monitoring unit 130's report generator 420 generates “reports on the data activity” and sends them to server utility 150. Bittmann, [0054], [0062], Fig. 4.

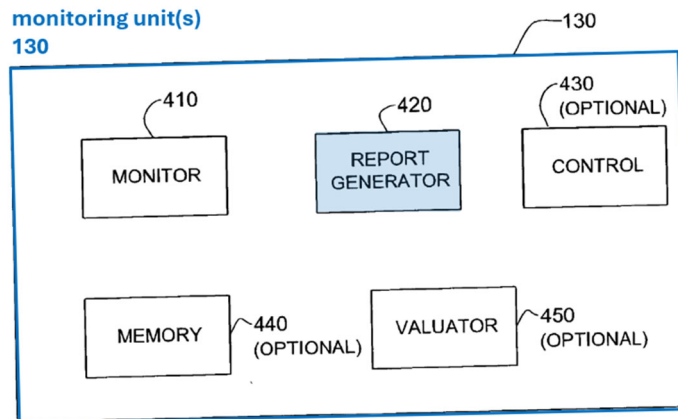


FIG. 4

Bittmann, Fig. 4.

202. The report may include, for example, the “volume of data transferred (e.g. kilobytes, number of packets, etc),” the “types of communication infrastructure utilized,” “the application on communication device 110/140 using the data,” and “any web sites accessed.” Bittmann, [0060]. Those fields describe the device’s use of data services: how much data the device transferred, over what infrastructure, which application used it, and which web sites the device accessed.

203. Because Bittmann permits monitoring unit 130 to monitor only some of a device’s data activity, in the combination the report would cover the device’s activity with Wright’s network, which is the network traffic that the server and probe reports would also cover, so that the reports can be compared. Bittmann explains that “in some embodiments this monitoring unit 130 monitors all data activity of

device 110, while in other embodiments, some data activity of device 110 may not be monitored,” and that the monitored activity “includes some or all of the data transferred to and/or from device 110 over communication network 120.” Bittmann, [0031]. As explained in the motivation-to-combine section of this ground, monitoring unit 130 in the combination would monitor the device’s activity with Wright’s network. It would report the traffic the device exchanges with that network, including the “emails” and “voice over IP” traffic that Bittmann lists, because that is the traffic for which the enterprise’s servers and probe would produce independent reports, and a comparison requires that both reports cover the same traffic. *Id.*

204. Accordingly, in my opinion, Wright-Bittmann teaches that system 200 receives monitoring unit 130’s report (*receiving a device data usage report from the end-user device*). Bittmann, [0031], [0054], [0060], [0062].

b. 16[b]: “obtaining a trusted data usage report associated with the end-user device; and”

205. In my opinion, Wright-Bittmann teaches 16[b].

206. In the combination, system 200 would receive a report of the device’s data activity either from an enterprise server, such as the corporate mail server or a VoIP server on the network, as explained for 15[a], or from a probe on Wright’s network. Bittmann, [0042]–[0043]; Wright, [0161], [0201]–[0202]. Bittmann identifies probes, like VoIP servers, as system resources whose reports of “data

activity” server utility 150 rates, and mediation system 210 receives “other reports of activity” in addition to monitoring unit 130’s reports. Bittmann, [0042]–[0043]. A probe on Wright’s network would observe the device’s traffic as that traffic crosses the network and would report the activity it observes to mediation system 210, as explained in the motivation-to-combine section of this ground.

207. Either report would cover traffic in the session that the device opened with its token, so it would be associated with the device. As explained in the motivation-to-combine section of this ground, client 201 opens each session with its encrypted token, which identifies the device’s client management record, and system 200 would match each server or probe report to the session whose traffic it covers, using the device’s network address and the time interval that the report identifies, and through the session’s token to the device’s record (*associated with the end-user device*). Wright, [0078], [0104], [0128], [0133]; Bittmann, [0060].

208. The device’s report would overlap with either independent report. As explained for 16[a], monitoring unit 130’s report would cover the device’s data activity with Wright’s network, which includes “emails” and “voice over IP.” Bittmann, [0031]. The corporate mail server handles the device’s email, a VoIP server on the network would handle its IP telephony, and a probe on Wright’s network would observe the device’s traffic with that network. Therefore, the activity

the device reports would overlap with the data activity that either the server or the probe reports.

209. In my opinion, a POSITA would have been motivated to obtain either the server report or the probe report, for the reasons explained in the motivation-to-combine section of this ground. Either report would let the administrator track and limit the device's usage, which is the first reason, and either would give system 200 an independent measurement against which to check the device's report, which is the second reason. Bittmann identifies both kinds of source, "probes" and "VoIP servers," among the system resources whose reports server utility 150 uses. Bittmann, [0043].

210. The two reports differ in scope. The server report would target the device's use of that server's service, for example the email data the corporate mail server exchanged with the device, while the probe report would cover the device's network traffic more broadly, across the services the device uses on Wright's network. Either report would provide an independent measurement of the traffic it covers, and either would be comparable with the overlapping portion of the device's report.

211. Either report, from the server or from the probe, would be trusted because the enterprise's own resources would generate it and the device would not be able to alter it. The server would generate its report from its own handling of the

device's traffic, and the probe would generate its report from its own observation of the traffic on the network; neither report would depend on what client 201 reports. Both resources would operate on the enterprise's network, outside the mobile device, as part of the infrastructure that "belong[s]" to the enterprise as network operator. Bittmann, [0035], [0042]–[0043]. A user who tampered with client 201 or with monitoring unit 130's reports would have no way to change the server's or the probe's measurements. The '777 Patent likewise describes a trusted source that "is a network-based element such as a server, gateway, proxy or router" and that "measures the service usage, and provides a service usage classification report back to a service controller so that it can be reconciled." '777 Patent, 64:37–45.

212. Accordingly, in my opinion, Wright-Bittmann teaches obtaining either an enterprise server or probe report of the device's activity in its token-opened session (*obtaining a trusted data usage report associated with the end-user device*). Bittmann, [0042]–[0043]; Wright, [0133], [0161], [0201]–[0202]; '777 Patent, 64:37–45.

c. 16[c]: "reconciling the device data usage report and the trusted data usage report."

213. In my opinion, Wright-Bittmann teaches 16[c].

214. In the combination, mediation system 210 would receive two reports concerning the mobile device's activity: monitoring unit 130's report (*the device*

data usage report), as explained for 16[a], and the server or probe report (*the trusted data usage report*), as explained for 16[b]. Mediation system 210 correlates and aggregates “reports from multiple sources to produce a unified report record,” which is “formatted to a protocol known to a rating system 220” and which rating system 220 then rates. Bittmann, [0042]–[0043].

215. To correlate the device report with the server or probe report, mediation system 210 would match the reports by the token-opened session whose traffic each covers, as explained for 16[a] and 16[b], and compare the usage, such as the data volume, that each reports for the same traffic. In my opinion, a POSITA would have understood that correlating reports from multiple sources concerning the same activity requires these operations: identifying which reports describe the same activity and comparing what each says about it. Bittmann, [0042], [0060].

216. A difference between the reports would show that client 201 misreported its usage. For overlapping traffic in the same session, the server or probe would measure what actually passed through the server or crossed the network, so a device report that shows less volume than the independent report, or omits a transfer the independent report shows, would reveal that client 201 understated its activity. As explained in the motivation-to-combine section of this ground, such misreporting indicates a compromise that authorization and authentication at the start of the session do not detect. Wright, [0052], [0255]. Mediation system 210 would then

combine the device report with the server or probe report into a unified report record for rating, as Bittmann describes. Bittmann, [0042]–[0043].

217. A POSITA would have had a reasonable expectation of success because mediation system 210 already receives, correlates, and reformats reports from multiple sources for rating, and the device report and the independent report would identify the same session and the same usage quantities, such as the volume of data transferred and the time interval, which is the information mediation system 210 needs to match and compare them. Bittmann, [0042]–[0043], [0060], [0069].

218. Accordingly, in my opinion, Wright-Bittmann teaches that mediation system 210 matches, compares, and combines the device report with either the server or probe report (*reconciling the device data usage report and the trusted data usage report*). Bittmann, [0042]–[0043], [0060].

5. Claim 17: “The method recited in claim 16, wherein obtaining the trusted data usage report comprises obtaining the trusted data usage report from a network element.”

219. In my opinion, Wright-Bittmann teaches claim 17.

220. As explained for 16[b], system 200 would obtain *the trusted data usage report* from an enterprise server or a probe on Wright’s network. Bittmann treats probes and VoIP servers as “system resources” that report data activity: rating system 220 rates “data activity reported by system resources (for example probes, switches, VoIP servers, etc).” Bittmann, [0043]. Bittmann’s list is open-ended

(“etc”), and a corporate mail server that handles the device’s email traffic is the same kind of resource as the VoIP servers it names. Each is a component of the network that carries or serves the device’s traffic: the probe would sit on the network where the device’s traffic passes, and the server is a node on the enterprise’s network that exchanges traffic with the device as it provides its service. In my opinion, a POSITA would have understood each to be a network element. The ’777 Patent likewise identifies “a server, gateway, proxy or router” as examples of a “network-based element.” ’777 Patent, 64:37–45.

221. Accordingly, in my opinion, Wright-Bittmann teaches that system 200 obtains the server or probe report (*obtaining the trusted data usage report*) from the enterprise server or the probe on Wright’s network (*from a network element*). Bittmann, [0042]–[0043]; ’777 Patent, 64:37–45.

6. Claim 19: “The method recited in claim 16, wherein obtaining the trusted data usage report comprises obtaining the trusted data usage report from a good customer feedback source.”

222. In my opinion, Wright-Bittmann teaches claim 19.

223. As explained for 16[b], system 200 would obtain *the trusted data usage report* from either an enterprise server or a probe, and a POSITA would have been motivated to obtain either. In my opinion, it would have been obvious for system 200 to obtain the trusted data usage report from an enterprise server providing the

service, such as the corporate mail server or a VoIP server. The server's report would be associated with the device's token, and, as explained for 15[a], such a server is a *good customer feedback source*: it is "a server or website that provides the service," and it would report the device's interaction with that service in association with the device credential. '777 Patent, 65:17–32. Claim 16's steps would then proceed with the server's report as the trusted report; mediation system 210 would reconcile monitoring unit 130's report with it, as explained for 16[c].

224. Accordingly, in my opinion, Wright-Bittmann teaches that system 200 obtains the enterprise server's report of the device's activity (*obtaining the trusted data usage report*) from the corporate mail server or VoIP server that provides the service (*from a good customer feedback source*). Bittmann, [0042]–[0043]; Wright, [0161], [0201]–[0202]; '777 Patent, 65:17–32.

7. **Claim 20: "The method recited in claim 19, wherein the good customer feedback source comprises a website or server that tracks a number of visits by the end-user device, a number of transactions associated with the end-user device, an amount of business generated in association with the end-user device, an amount of data communicated by or to the end-user device, a measure of an interaction by the end-user device with the website or server, or a combination of these."**

225. In my opinion, Wright-Bittmann teaches claim 20.

226. As explained for 16[b] and claim 19, the *good customer feedback source* is an enterprise server, such as the corporate mail server or a VoIP server

(*comprises a ... server*). The server would report the device's "data activity." Bittmann, [0043]. For mediation system 210 to correlate that report with the device's report, the server's report would include the same usage quantities, such as the volume of data transferred, as explained for 15[a] and 16[c]. *Id.*, [0060]. To report that volume, the server would count the data it exchanges with the device as it provides its service; a mail server, for example, would count the data in the messages it delivers to and accepts from the device. The server therefore would track the amount of data the device communicates through it (*tracks ... an amount of data communicated by or to the end-user device*).

227. Accordingly, in my opinion, Wright-Bittmann teaches that the enterprise server (*the good customer feedback source comprises a ... server*) counts the data it exchanges with the mobile device (*tracks ... an amount of data communicated by or to the end-user device*), an alternative expressly recited in claim 20. Bittmann, [0043], [0060].

8. Claim 21: "The method recited in claim 16, wherein the trusted data usage report comprises a flow data record."

228. In my opinion, Wright-Bittmann teaches claim 21.

229. The '777 Patent describes flow data records as "detailed reports of service usage for each service flow, such as network socket connection, opened and used to transmit data to or from the device." '777 Patent, 39:26–40:32. When a

service flow closes, a record “indicating source address, destination address and amount of data transmitted” is created. *Id.* The ’777 Patent also identifies a network element as a source of flow data records. *Id.*, 64:5–11.

230. In the combination, *the trusted data usage report* for this claim would be the probe’s report, as explained for 16[b]. Bittmann does not describe the format of a probe’s report. The device’s report, however, may include the “volume of data transferred,” the “time-interval for data transfer,” the “source and destination of data,” and “the application protocol.” Bittmann, [0060]. Bittmann’s monitor 410 categorizes “a specific data activity” and passes “pre-determined information on the activity” to report generator 420. *Id.*, [0059]. In my opinion, these fields describe an individual transfer: its endpoints (source and destination), its size (volume), its timing (time interval), and its protocol.

231. In my opinion, comparing aggregate usage would show that the device misreported its usage, as explained for 16[c], but not which transfer was misreported, when, or to what destination. A total for a session or reporting interval combines all of the device’s transfers. If the device’s total falls short of the probe’s total, the difference shows that the device understated its activity, but it does not identify the transfer the device omitted or understated, the time of that transfer, or the destination to which the data went.

232. In my opinion, to identify additional information about the activity of a device suspected of compromise, a POSITA would have been motivated to have the probe report the same fields as the device for each transfer it observes, so that mediation system 210 would be able to match each record and assess the compromised device's activity. Wright is concerned with "unauthorized or rogue mobile devices trying to hijack the security policies or corporate data." Wright, [0052]. Knowing which transfer a compromised client misreported, when it occurred, and where the data went would tell the administrator what corporate data left the network and to what destination. Bittmann identifies its reports as useful "for fraud and network management." Bittmann, [0059]. With per-transfer records from both sources, mediation system 210 would pair each probe record with the device's report of the same transfer by its source, destination, protocol, and interval, and identify any transfer the device omitted or misstated.

233. Doing so would have required only configuring the probe to report fields that Bittmann specifies for the device's report. In the combination, the probe would already observe the device's traffic and report data activity to mediation system 210, and each packet it observes carries its source and destination addresses and a protocol identifier; grouping packets by those values and totaling their size over each transfer uses information the probe already sees. Bittmann, [0042]–[0043],

[0060]. The results would therefore have been predictable, and in my opinion a POSITA would have had a reasonable expectation of success.

234. The probe would group the packets of each transfer it observes, that is, the packets sharing a source, destination, and protocol during the transfer, measure their volume, and send a record for each transfer to mediation system 210, which receives “other reports of activity” in addition to the device’s reports. Bittmann, [0042]–[0043], [0060]. Each such record would report a single transfer’s endpoints, protocol, and data amount. In my opinion, that is a report of service usage for a service flow, which is how the ’777 Patent describes a flow data record: a report “of service usage for each service flow, such as network socket connection,” indicating “source address, destination address and amount of data transmitted.” ’777 Patent, 39:26–40:32.

235. Accordingly, in my opinion, Wright-Bittmann teaches a probe data report of each transfer’s source, destination, protocol, data volume, and interval (*trusted data usage report comprises a flow data record*). Bittmann, [0042]–[0043], [0059]–[0060]; ’777 Patent, 39:26–40:32.

9. Claim 22: “The method recited in claim 16, further comprising adjusting a user account associated with the end-user device based on reconciling the device data usage report and the trusted data usage report.”

236. In my opinion, Wright-Bittmann teaches claim 22.

237. As explained for 16[c], mediation system 210 would combine the reconciled reports into a unified report record. That record would be “formatted to a protocol known to a rating system 220,” which would rate it. Bittmann, [0042]. Allowance manager 230 then “updates the allowance(s) for the account associated with communication device 110 ... based on the rating of activity involving communication device 110.” *Id.*, [0043]. In my opinion, because the rating would be of the reconciled record, the update would be based on the reconciliation: the account would change by the rated amount of the activity as the reconciled record states it.

238. System 200 would identify the account through the client management record of the session the device opened with its token, as explained for 15[b]. Wright, [0104]–[0105]. In my opinion, a POSITA would have been motivated to do so for the first reason explained in the motivation-to-combine section of this ground: to let the administrator monitor and limit how much each device uses a permitted service, system 200 must apply the reconciled usage to the account of the device that used it, and the client management record is how system 200 already identifies each managed device and its user. *Id.*

239. Accordingly, in my opinion, Wright-Bittmann teaches that allowance manager 230 updates the allowance (*adjusting a user account*) of the account associated with the mobile device (*associated with the end-user device*) using the

rating of the unified record that mediation system 210 produces by matching, comparing, and combining (*based on reconciling*) monitoring unit 130's report (*the device data usage report*) and the server or probe report (*and the trusted data usage report*). Bittmann, [0042]–[0043]; Wright, [0104]–[0105].

- 10. Claim 28: “The method recited in claim 16, wherein the device data usage report indicates usage associated with a classification, the classification being one or more device applications, one or more network destinations, one or more network types, one or more sponsored services, one or more background activities, one or more operating system (OS) programs, one or more OS libraries, one or more OS functions, or a combination of these.”**

240. In my opinion, Wright-Bittmann teaches claim 28.

241. As explained for 16[a], monitoring unit 130's report is *the device data usage report*. Monitor 410 “may be configured to categorize a specific data activity” according to, among other things, “the source or destination of the data transferred to or from device 110/140” and “the application on communication device 110/140 using the data,” and, depending on the category, it transfers “certain pre-determined information on the activity” to report generator 420 for transmission to server utility 150. Bittmann, [0059]. The report may include, for a data activity, the “volume of data transferred,” the “source and destination of data,” and “the application on communication device 110/140 using the data.” *Id.*, [0060]. In my opinion, a POSITA would have understood that the report thus attributes an activity's data

volume to the application that used the data and to the data's destination, so that the reported usage is associated with a classification by application and by destination.

242. Accordingly, in my opinion, Wright-Bittmann teaches that monitoring unit 130's report (*device data usage report*) attributes an activity's data volume to an application and a destination (*indicates usage associated with a classification ... being one or more device applications ... one or more network destinations*), which are alternatives expressly recited in claim 28. Bittmann, [0059]–[0060].

11. **Claim 29: “The method recited in claim 16, wherein the trusted data usage report indicates usage associated with a classification, the classification being one or more device applications, one or more network destinations, one or more network types, one or more sponsored services, one or more background activities, one or more operating system (OS) programs, one or more OS libraries, one or more OS functions, or a combination of these.”**

243. In my opinion, Wright-Bittmann teaches claim 29.

244. As explained for 16[b], *the trusted data usage report* for this claim is the probe's report. To reconcile that report with the device's report, as explained for 16[c], the probe would record the same quantities that the device reports, including the source and destination of the traffic, as explained in the motivation-to-combine section of this ground. Bittmann, [0060].

245. As explained for claim 28, the device's report attributes data volume to destinations. In my opinion, a POSITA would have been motivated to have the probe

do the same, recording the destination address of the traffic it observes and accumulating the volume sent to each destination during the reporting interval, so that mediation system 210 would be able to compare usage by destination. Bittmann, [0042]–[0043], [0060].

246. That comparison would catch a device that reports the correct total but attributes it to the wrong destination. A compromised client that reports the true total volume of its traffic, but attributes part of that volume to a permitted destination such as the corporate mail server when the data actually went elsewhere, would pass a comparison of totals. A comparison by destination would show that the probe measured traffic to a destination the device did not report, or more traffic to a destination than the device reported. Wright, [0052].

247. A POSITA would have had a reasonable expectation of success because the probe would observe the addresses of the traffic it measures, Bittmann's reports already carry destination fields, and mediation system 210 already correlates reports. Bittmann, [0042]–[0043], [0060]. Each packet the probe observes carries its destination address, so accumulating volume by destination requires only a count kept for each address.

248. Accordingly, in my opinion, Wright-Bittmann teaches that the probe's report (*trusted data usage report*) would attribute data volumes to destinations (*indicates usage associated with a classification ... being ... one or more network*

destinations), an alternative expressly recited in claim 29. Bittmann, [0042]–[0043], [0060].

12. **Claim 31: “The method recited in claim 16, wherein the device data usage report comprises information identifying an application, an operating system (OS) function, an OS program, an OS library, a network resource, a web site, a network source, a network destination, a domain, a uniform resource locator (URL), an IP address, a port, a protocol, a socket tuple, a server, a network route, an access point name (APN), a gateway, a proxy, a content source, a sub-network, a quality-of-service (QoS) level, a content type, a background activity, or a combination of these.”**

249. In my opinion, Wright-Bittmann teaches claim 31.

250. As explained for 16[a], monitoring unit 130’s report is *the device data usage report*. The report may include “the application on communication device 110/140 using the data,” “any web sites accessed,” the “source and destination of data,” and “the application protocol.” Bittmann, [0060]. The report therefore includes information identifying the application using the data, the web sites accessed, the data’s source and destination, and the protocol.

251. Accordingly, in my opinion, Wright-Bittmann teaches that monitoring unit 130’s report (*device data usage report*) includes *information identifying* the application using the data (*an application*), the web sites accessed (*a web site*), the data’s source and destination (*a network source, a network destination*), and the

protocol (*a protocol*), each an alternative expressly recited in claim 31. Bittmann, [0060].

13. **Claim 32: “The method recited in claim 16, wherein the trusted data usage report comprises information identifying an application, an operating system (OS) function, an OS program, an OS library, a network resource, a web site, a network source, a network destination, a domain, a uniform resource locator (URL), an IP address, a port, a protocol, a socket tuple, a server, a network route, an access point name (APN), a gateway, a proxy, a content source, a sub-network, a quality-of-service (QoS) level, a content type, a background activity, or a combination of these.”**

252. In my opinion, Wright-Bittmann teaches claim 32.

253. As explained for 16[b], *the trusted data usage report* for this claim is the probe’s report. To reconcile that report with the device’s report, as explained for 16[c], the probe would record the same quantities that the device reports, including the source, destination, and protocol of the traffic, as explained in the motivation-to-combine section of this ground. Bittmann, [0060].

254. As explained for claim 31, the device’s report identifies each of these. In my opinion, a POSITA would have been motivated to have the probe identify them too, so that mediation system 210 would be able to compare usage by endpoint and protocol, not just in total. A comparison by endpoint and protocol would reveal a client that reports the correct total but attributes traffic to a different destination, as explained for claim 29, or to a different protocol. A POSITA would have had a

reasonable expectation of success because the probe would read the source and destination addresses and protocol identifiers from the headers of the packets it observes, and Bittmann's reports already carry source, destination, and protocol fields. Bittmann, [0042]–[0043], [0060].

255. Accordingly, in my opinion, Wright-Bittmann teaches that the probe's report (*trusted data usage report*) would identify the source, destination, and protocol of the traffic (*comprises information identifying ... a network source, a network destination ... a protocol*), alternatives expressly recited in claim 32. Bittmann, [0042]–[0043], [0060].

C. GROUND 1C: CLAIMS 8, 9, 12, AND 23–25 ARE OBVIOUS OVER WRIGHT AND DURHAM

1. Durham's Teachings

256. I described Durham in the Overview of Durham section above. The motivation to combine rests principally on three of Durham's teachings: the device-bound identity of firmware agent 125, established at enrollment; the agent's signed, nonce-bearing responses, which the network verifies; and the agent's firmware-enforced default filtering, which restricts the host to network access control exchanges. Durham, [0015], [0017], [0019], [0021], [0025], [0027]–[0029]. For 8[f] and claims 23–25, I also rely on Durham's restriction of hosts that circumvent its controls, its remediation instructions, its server-side detection of modified or

withheld measurements, and its defense against delayed messages, as I explain for those claims. *Id.*, [0026], [0030]–[0032], [0039], [0046].

257. A POSITA would have understood what these mechanisms establish for the network. A signature that verifies under firmware agent 125’s key shows that the report came from the enrolled firmware agent; Durham explains that the signature “provides assurance to a PDP component that the data is submitted by an entity possessing the pertinent keying material.” Durham, [0025]. The enrollment binding ties that firmware agent to a particular host platform 100. *Id.*, [0017]–[0019]. The fresh nonce shows that the report was produced for the current exchange rather than replayed from an earlier one, and Durham explains that mutual authentication with proof of liveness “may defeat replay and delayed messages attacks on the host platform.” *Id.*, [0025], [0027]–[0028], [0039].

258. In my opinion, Durham thus provides a device-side agent that the network can rely on even when it cannot rely on the host’s ordinary software. Firmware agent 125 has a device-bound identity, protected from host software, that enrollment binds to its host platform. Durham, [0012], [0017], [0019]. It answers the network’s nonce with a signed response that the network verifies before relying on it. *Id.*, [0025], [0027]–[0029]. And it controls traffic filters 120, to which it has exclusive access, so that by default the host can exchange only network access control messages until the network decides what access to permit. *Id.*, [0012],

[0015], [0021]. Because no code executing within host operating system 110 can tamper with firmware agent 125, prevent it from running, or circumvent it, these functions do not depend on the host's software being trustworthy, and the server can detect when a rogue trust agent modifies or withholds the agent's measurements. *Id.*, [0010], [0012]–[0013], [0026]. As explained in the next section, a POSITA would have been motivated to add firmware agent 125 to Wright's client 201 and to apply these techniques in Wright's system.

2. Motivation to Combine Wright and Durham

259. In my opinion, a POSITA would have been motivated to add Durham's firmware agent 125 to Wright's client 201 as an additional device agent, alongside policy setting module 212, policy enforcement control module 214, and authorization module 245, and to apply Durham's other techniques (principally its device-bound identities, signature verification, and firmware-enforced default filtering, and, where particular claims call for them, its remediation instructions, its restriction of hosts that circumvent its controls, its server-side detection of modified or withheld measurements, and its defense against delayed messages) to Wright's system. I explain below how the combined system would operate, the reasons a POSITA would have made the combination, and why a POSITA would have had a reasonable expectation of success. How each technique applies to the claims is explained for claims 8, 9, 12, and 23–25.

260. In the combination, system 200 would still authorize and authenticate client 201 as explained for 1[a]–1[d] in Ground 1A: client 201 sends its status request with the encrypted token, and system 200 compares client 201’s response to its random challenge with its own SHA1 hash. Wright, [0128]–[0129], [0133]–[0137]. Durham’s technique would be added after that step. After authenticating client 201 and before sending any policy (Wright, Fig. 3B, steps 328/327), system 200 would use firmware agent 125 to confirm that the device presenting the token is the device to which system 200 issued the token. *Id.*, [0128], [0138]. How system 200 would make that confirmation is explained for 8[d].

261. Wright recognizes the need to guard against rogue devices. Its authorization module 232, which authorizes a client’s communication exchanges with system 200, “is a further safeguard against unauthorized or rogue mobile devices trying to hijack the security policies or corporate data.” Wright, [0052]. Authorization module 232 decides whether to authorize an exchange using the authorization information client 201 sends with its status request, and, as I explained for 1[a]–1[d] in Ground 1A, a POSITA would have understood that system 200 authenticates client 201 in that exchange using client 201’s response to the Establish Session random challenge. *Id.*, [0052], [0128], [0133]–[0137].

262. That safeguard, however, rests on information held by client 201, which is software on the mobile device. Wright, [0046]. Client 201 holds the encrypted

token that system 200 issued to the device and sends it as authorization information with its status request. *Id.*, [0078], [0128]–[0129]. Client 201 also holds the enterprise’s key, which it uses to compute its response to the random challenge: when system 200 is installed, “a key is generated and stored in an encrypted file,” and that file “is copied and distributed to the clients.” *Id.*, [0130], [0136]. Both credentials on which authorization module 232 relies are therefore data stored with client 201’s software on the device.

263. In my opinion, a device running a copy of client 201, with its stored token and key file, would therefore pass Wright’s authorization and authentication. It would send the same encrypted token with its status request, and it would compute the correct “SHA1(random challenge, key)” response to system 200’s challenge, because it holds the same key. Wright, [0128]–[0130], [0135]–[0137]. The token identifies the device to which system 200 issued it, and a correct challenge response shows that the responding client is a managed client holding the enterprise’s key, as I explained for 1[a] and 1[d] in Ground 1A. *Id.*, [0078], [0130], [0137]. But both are data stored with client 201’s software, and a copy of that software carries both. *Id.*, [0078], [0128]–[0130]. System 200 thus would have no way to tell that the device presenting the token was not the device to which it issued the token.

264. Durham addresses that problem. In my opinion, a POSITA would have been motivated to apply Durham’s techniques to make Wright’s client-server system

more resilient to rogue or compromised mobile devices. Durham is directed to enforcing network access policies “even when the host operating system is compromised,” and its firmware agent 125 provides “tamper resistant posture reporting and remediation” that host software cannot tamper with, prevent from running, or circumvent. Durham, [0010], [0012]. Those properties address the weakness in Wright’s safeguard just described: Wright’s credentials can be copied along with client 201’s software, whereas Durham’s protections do not depend on that software. *Id.*; Wright, [0052], [0130].

265. First, Durham’s enrollment would give Wright a device-bound identity. Durham’s enrollment may “cryptographically bind the identities of host platform 100 with firmware agent 125,” and those identities may be “protected using Trusted Platform Module (TPM) 130.” Durham, [0017]. The firmware agent’s identity is established “using Public-Key/Private-Key Identity Certificates,” and the agent signs its responses with its own private key. *Id.*, [0019], [0025]. A POSITA would have understood that, unlike Wright’s token and enterprise key, that private key is not data that a copy of client 201’s software would carry: it is the key of firmware agent 125, which host code cannot tamper with; it belongs to an identity protected using the TPM; and it is bound to the enrolled host platform. *Id.*, [0012], [0017], [0019]. A POSITA therefore would have understood that Durham’s device-bound

identities would have provided a way for Wright to identify devices with copied client software and to restrict their network access.

266. Second, Durham's enforcement techniques are resilient to host compromise. Firmware agent 125 enforces policies "even when the host operating system is compromised." Durham, [0010]. By default, "filters 120 block all access by host system 100, but firmware agent 125 may enable only network access control exchanges," so a host that the network has not verified can only negotiate with the network. *Id.*, [0021]. Third, Durham lets the server detect tampering: "if a rogue trust agent modifies or does not send firmware agent 125 integrity measurements, this action can be detected at the server," and the mutual authentication with proof-of-liveness between firmware agent 125 and the PDP "may defeat replay and delayed messages attacks on the host platform." *Id.*, [0026], [0039]. Each of these techniques would make Wright's system more resilient to rogue or compromised mobile devices, because each continues to work when the device's ordinary software, including client 201, cannot be trusted.

267. In my opinion, the combination would have applied known techniques (Durham's device-bound identities, signature verification, and firmware-enforced default filtering) to improve a similar system (Wright's client-server system, which authenticates managed mobile devices and distributes security policies to them) in the same way, with the predictable result that system 200 would identify devices

running copied clients and that client devices would be more resilient to compromise. Durham, [0010], [0012], [0017], [0019], [0021], [0025]–[0029]; Wright, [0046]–[0048], [0052], [0128]–[0129], [0133]–[0137]. Wright’s system is similar to Durham’s: in each, a network-side server decides whether to admit a device after an authenticated exchange with software on the device, and then sends policy that device-side agents enforce. Wright, [0128], [0133]–[0138]; Durham, [0015]–[0016], [0027]–[0032]. In Durham, those techniques give a network assurance about a host whose operating system is compromised; in the combination, each technique would perform the same function for the mobile devices that system 200 manages.

268. In my opinion, a POSITA would have had a reasonable expectation of success in making the combination. Adding firmware agent 125 to client 201 would have been consistent with Wright’s design, because Wright explains that its modules “can be implemented as software, hardware, firmware or any combination of the three.” Wright, [0286]. Client 201 thus need not be software alone, and a firmware agent operating alongside client 201’s software modules fits Wright’s own description of how its modules may be built. Durham’s firmware agent 125 likewise operates on the same host platform as the host’s software agents: it communicates with the host system through system interface 305 and with trust agent 105 over an authenticated intra-platform channel, and it has its own memory, storage, and control

circuitry, which let it “perform operations independent of the host platform and/or host operating system.” Durham, [0012], [0024], [0048]–[0049]. And Wright’s mobile devices include laptop computers such as laptop 148, each a computing device with a processor coupled to one or more network interfaces, which is the same kind of networked computer platform that Durham describes, with traffic filters on a network interface card that firmware agent 125 controls. Wright, [0042], [0046]; Durham, [0011]–[0012].

269. Further, Wright’s session protocols “are designed to be extensible and straightforward to allow addition of new verbs later.” Wright, [0133]. Durham’s exchange (the network’s nonce, the agent’s signed response, and the network’s verification) is a request-and-response exchange between a device-side agent and a network-side decision point, the same kind of exchange that Wright’s “Establish Session” protocol and its “Query Policy Update” and “Download Policy” verbs carry between client 201 and system 200. *Id.*, [0133]–[0147]; Durham, [0025]–[0029]. A POSITA would have understood how to carry Durham’s exchange as a new verb in Wright’s session and how to apply Durham’s other techniques to Wright, as I explain for each claim below. Durham’s techniques also are not limited to a particular kind of device: its host platform “is intended to represent a broad range of electronic devices that may be coupled with a network.” Durham, [0011]. They therefore apply to the mobile computing devices on which Wright’s client 201 runs. Wright, [0046].

270. Implementing the combination would have required adding firmware agent 125 to client 201; enrolling the agent when system 200 issues the device's token; and programming system 200 to send a nonce, verify the agent's signed response with the public key in the stored Identity Certificate, and, when that check fails, send remediation instructions and direct the agent to apply its default configuration. Durham describes each of those pieces: enrollment with Public-Key/Private-Key Identity Certificates; the PDP's nonce, the agent's signed response, and digital signature verification; the return of remediation instructions; and the agent's default configuration. Durham, [0017], [0019], [0021], [0025]–[0031], [0046]. Wright, for its part, issues the token during a trusted connection, keeps per-device client management information in which the certificate would be stored, and designs its protocols to be extended. Wright, [0078], [0104], [0133].

3. Claim 8

- a. **8[a]: “The method recited in claim 1, wherein the device credential is a first device credential, and wherein the end-user device is a first end-user device, and wherein the service policy is a first access network control policy, and further comprising:”**

271. In my opinion, Wright-Durham teaches 8[a].

272. Claim 8 depends from claim 1 and adds steps that occur at different times. Claim 1 is performed in a session (a “first session”) in which a first mobile device presents its encrypted token and system 200 executes the authorization and

authentication procedure explained for claim 1 in Ground 1A. Wright, [0128]–[0129], [0133]–[0137]. 8[c]–8[f] apply when the same device, or another device, later presents a token in its own session. 8[b], the association of the token with the first mobile device, occurs earlier, when system 200 issues the token and before the first mobile device presents it in 1[a]. That association is what allows system 200 to use the token to obtain the device’s device-level policy in 1[b], as explained for 1[b] in Ground 1A. *Id.*, [0078], [0104], [0128].

273. As to 8[a], in the first session, system 200 receives the first mobile device’s (*first end-user device*) encrypted token (*device credential is a first device credential*) with the device’s status request and obtains the *service policy* applicable to the device, as explained for 1[a] and 1[b] in Ground 1A. Wright, [0078], [0104], [0128]–[0129].

274. In my opinion, the *service policy* is a *first access network control policy* because both of its portions control the device’s network access. Its first portion is the device’s security rules, which system 200 sends to client 201 and client 201 enforces, and those rules control network access: Wright’s policies specify, for example, “ports to be monitored,” “network services to be monitored,” “applications to be monitored,” and “enforcement mechanisms to be put in place.” Wright, [0048]. A security policy “may not allow a particular network service or application or both to be used.” *Id.*, [0014]. Its second portion denies the device access to system 200:

when the client's response does not match, system 200 "drops the connection with no further ado." *Id.*, [0137]. As explained for 1[b] in Ground 1A, both portions belong to the device's *service policy*. A POSITA would have understood a policy whose rules monitor the device's ports, network services, and applications, specify the enforcement mechanisms to be put in place, and can disallow particular network services or applications, and that determines whether the device may communicate with system 200 at all, to be a policy that controls the device's access to the network.

275. Accordingly, Wright-Durham teaches that, in the first session, system 200 receives the first mobile device's (*first end-user device*) encrypted token (*device credential is a first device credential*) and obtains the policy applicable to that device (*service policy*), whose security rules and denial of access both control the device's network access (*first access network control policy*). Wright, [0014], [0048], [0078], [0104], [0128]–[0129], [0137].

b. 8[b]: "associating the first device credential with the first end-user device;"

276. In my opinion, Wright-Durham teaches 8[b].

277. Wright's policy management module 236 "designates 304 one or more client devices associated with the policy" and, in a secure manner, provides 306 each designated device with authorization information "for use in contacting the server system securely." Wright, [0078]. "One example of authorization information is an

encrypted token provided by the authorization module 232 to the mobile client device during a trusted connection between the two,” such as an internal connection behind the enterprise’s firewall. *Id.*

278. System 200 also “maintains 314 client management information for the mobile device and the one or more policies associated with it,” organized in a data object accessible to system 200. Wright, [0104], [0120].

279. A POSITA would have understood that system 200 links the token it issues to the device’s client management information, so that when the token is later presented, system 200 can identify the device to which it issued the token and that device’s device-level policy, as explained for 1[b] in Ground 1A. Wright, [0078], [0104], [0128]. System 200 issues the token to a particular device during a trusted connection and keeps a management record specific to that device; linking the two is how system 200 knows to which device it issued which token. *Id.*, [0078], [0104]. By linking the token to the first mobile device’s client management information, system 200 *associat[es]* the token (*first device credential*) with the first mobile device (*first end-user device*), as described for 8[a].

280. In the combination, when system 200 provides the token to the device during the trusted connection, it would also enroll the device’s firmware agent 125, which the combination incorporates into client 201. Durham’s enrollment establishes the agent’s identity “using Public-Key/Private-Key Identity Certificates”

and binds that identity to the host platform. Durham, [0017], [0019]. The trusted connection is the setting in which system 200 already gives the device its credential, so enrolling the agent there ensures that the identity system 200 records is that of the device that actually receives the token. Wright, [0078]. Durham's network side likewise holds the credential it needs to verify the agent: Durham relies on "the security association between network access PDP 150 and firmware agent 125" and on "associated security credentials" at the server, and its trust server verifies the agent's signed data "using digital signature verification." Durham, [0026], [0029]. In the combination, the credential that system 200 obtains at enrollment is the agent's Identity Certificate.

281. System 200 would store the agent's Identity Certificate with the token in the device's client management information. Wright, [0078], [0104]. In my opinion, a POSITA would have been motivated to do so because system 200 later needs to know which certificate belongs with which token: as explained for 8[d], system 200 would verify a device's signed response with the public key in the certificate stored with the token the device presents. Because a copy of client 201 carries the token, the token by itself does not show that the presenting device is the device to which system 200 issued it, while Wright's purpose is to safeguard against "unauthorized or rogue mobile devices trying to hijack the security policies or corporate data." Wright, [0052]. Storing the certificate in the per-device record in

which system 200 already keeps the device's policy and versions keeps the device's credentials together. *Id.*, [0104].

282. A POSITA would have had a reasonable expectation of success in storing the certificate with the token. Wright's client management information is already a per-device record that system 200 keeps for each mobile device, organized in a data object accessible to system 200. Wright, [0104], [0120]. Durham's Identity Certificate is data, carrying identifying information such as serial numbers and identifying strings, which system 200 would store as an additional field of the same record. Durham, [0019].

283. As a result, the token that system 200 issued is associated with a device-bound identity: the Identity Certificate of the firmware agent 125 enrolled on, and bound to, the first mobile device. Durham, [0017], [0019]; Wright, [0104]. System 200 uses that stored certificate in 8[d] to determine whether a device that presents the token is the first mobile device.

284. Accordingly, Wright-Durham teaches that system 200 links (*associating*) the encrypted token it issued (*the first device credential*) to the client management information of the first mobile device (*with the first end-user device*), in which system 200 also stores the Identity Certificate of that device's firmware agent 125. Wright, [0078], [0104]; Durham, [0017], [0019].

c. 8[c]: “determining whether a second device credential, associated with an authenticating end-user device, is the first device credential;”

285. In my opinion, Wright-Durham teaches 8[c].

286. When the first mobile device, or a different device (*an authenticating end-user device*), later activates a network connection in its own session, as described for 8[a], its client 201 would send system 200 a status request that provides its token (*second device credential*), as explained for 1[a] in Ground 1A. Wright, [0128]–[0129]. System 200 receives the request at step 322 and uses the presented token to identify the client management information and policy of the device to which it issued that token, as explained for 1[b] in Ground 1A. *Id.*, [0128] (steps 322–324). In doing so, system 200 *determin[es] whether* the presented token (*second device credential, associated with an authenticating end-user device*) is the token associated with the first mobile device under 8[b] (*is the first device credential*).

287. A POSITA would have understood that this lookup answers the question 8[c] asks. If the presented token is the token that system 200 issued to the first mobile device, the lookup returns the first mobile device’s client management information and policy. If it is not, the presented token is either another managed device’s token, in which case the lookup returns that other device’s information, or

a token that system 200 does not recognize, in which case the lookup returns no device. Wright, [0078], [0104], [0128].

288. Accordingly, Wright-Durham teaches that system 200 *determin[es] whether* the token presented in a later session (*a second device credential*) by the device then activating a connection (*associated with an authenticating end-user device*) is the token associated with the first mobile device (*is the first device credential*), which system 200 does by using the presented token to identify the client management information of the device to which it issued that token. Wright, [0078], [0104], [0128]–[0129].

d. 8[d]: “determining whether the authenticating end-user device is the first end-user device;”

289. In my opinion, Wright-Durham teaches 8[d].

290. 8[c] determines whether the presented token is the first token, but not whether the *authenticating end-user device* is the first mobile device. A device running a copy of client 201 holds both the first mobile device’s token and the enterprise’s key, so it presents the same token and passes the same authorization and authentication described for 1[d] in Ground 1A. Wright, [0128]–[0130], [0135]–[0137]. As explained in the motivation-to-combine section of this ground, Wright alone gives system 200 no way to tell that device apart from the first mobile device.

291. Durham supplies a device-bound identity. Firmware agent 125's identity is established "using Public-Key/Private-Key Identity Certificates," is bound to host platform 100 through an attribute certificate, and is "protected using Trusted Platform Module (TPM) 130." Durham, [0017], [0019]. A POSITA would have understood how such a certificate works: it binds a public key to the agent's identity, and the agent holds the corresponding private key. A signature made with the private key verifies with the certificate's public key, and without the private key it is computationally infeasible to produce a signature that does. Only the holder of the certificate's private key therefore can make a signature that verifies with the certificate's public key. Durham itself explains that a signature made with the agent's key "provides assurance to a PDP component that the data is submitted by an entity possessing the pertinent keying material." Durham, [0025].

292. In my opinion, firmware agent 125's private key also would not travel with a copy of client 201's software. Firmware agent 125 "may serve as a tamper-resistant execution environment on host platform 100," and in the combination it would serve as one that host code cannot tamper with, prevent from running, or circumvent; it has its own non-volatile storage and control circuitry and can "perform operations independent of the host platform and/or host operating system"; and its cryptographic identity may be "further protected using Trusted Platform Module (TPM) 130." Durham, [0012]–[0013], [0017], [0049]. A POSITA would

have understood that a private key held in that protected environment, as part of a TPM-protected identity, cannot be read by host software, so neither client 201 nor a copy of it can extract the key. And unlike Wright's token, which client 201 sends to system 200 with its status request, the private key is not sent to the network: firmware agent 125 sends signatures made with the key, not the key itself. *Id.*, [0025]; Wright, [0128]–[0129].

293. In my opinion, a POSITA would have found it obvious to add Durham's exchange to Wright's session to identify whether the *authenticating end-user device* is the first mobile device or a different device. After system 200 authorizes and authenticates the authenticating device's client 201, as explained for 1[d] in Ground 1A, and before it sends any policy at steps 328/327 of Wright's Figure 3B, system 200 would add Durham's exchange as a new verb in the established session. Wright, [0128], [0133], [0138]. Wright's session already sequences verbs in this way: once the session is established, client 201 sends an update query, and any policy is downloaded only afterward. *Id.*, [0138]–[0147]. Wright's protocols are "designed to be extensible and straightforward to allow addition of new verbs later." *Id.*, [0133].

294. Durham orders its exchange the same way. In Durham, host trust agent 105 initiates a network access request that establishes "a secure data channel for further communications" with network access PDP 150 "by employing any number of standards or proprietary protocols for authentication of each side," and PDP 150

sends its nonce in those further communications. Durham, [0027]. Meanwhile, firmware agent 125's default configuration permits network access control exchanges only: by default, filters 120 "block all access by host system 100," but firmware agent 125 "may enable only network access control exchanges." *Id.*, [0021]. Durham's firmware agent sets network filters to this "predefined default policy" upon a power-on or network link change event, and the default applies "until an acceptable security policy can be verified." *Id.*, [0040]. The network access control exchanges are not limited to layer-2 messages: the default may be set "by network administration policy when host platform 100 is configured for enterprise usage," and Durham's default policies may supply filter configurations for "L3 Virtual Private Network (VPN), L4 VPN, [and] Transport Layer Security (TLS) web server" connections "that may enable posture assessment." *Id.*, [0021], [0023]. In the combination, the network access control exchanges that firmware agent 125's default permits are Wright's status request, its "Establish Session" protocol, and the added verb: they are the exchanges by which the device negotiates its access with system 200 and receives system 200's decision. While the default is in place, the device can only negotiate with the network, not reach network resources, as I explain for claim 12. Wright, [0128]–[0129], [0133]–[0137]; Durham, [0021], [0023], [0040].

295. In the added exchange, system 200 would send the device a nonce, as Durham's PDP 150 does. Durham, [0027]. Firmware agent 125 would return a response that includes system 200's nonce and is signed with its private key, as Durham's agent does for PDP 150. *Id.*, [0025], [0028]. System 200 would then verify the signature using the public key in the Identity Certificate stored with the presented token in the device's client management information, as explained for 8[b], just as Durham's trust server 155 evaluates reported information "first for integrity and authenticity of the data, using digital signature verification." *Id.*, [0029]; Wright, [0104]. System 200 would also confirm that the returned nonce is the one it issued for this attempt. Durham, [0027]–[0028], [0043].

296. Because system 200 would issue a new, unpredictable nonce with each attempt, a recorded response would fail if replayed. Durham's PDP 150 sends "an unpredictable (random) nonce session identifier" "to establish proof-of-liveness for firmware agent 125," and firmware agent 125 includes that nonce in the digital signature it returns. Durham, [0027]–[0028]. Durham uses the same technique, a nonce included in the signed data, to "ensure that an intermediary rogue trust agent cannot capture and replay the measurements." *Id.*, [0025]. A response recorded during the first mobile device's earlier session would carry an earlier nonce and would not match the nonce system 200 issued for the current attempt. Durham

explains that this proof of liveness “may defeat replay and delayed messages attacks on the host platform.” *Id.*, [0039].

297. In my opinion, a verified signature shows that the response came from the device to which system 200 issued the token, and a failed verification shows that the response was not produced by the firmware agent enrolled on that device, and thus that it came from a different device. The certificate stored with the token belongs to the firmware agent 125 that system 200 enrolled on the first mobile device when it issued that token, as explained for 8[b], and only that agent holds the matching private key. Durham, [0017], [0019], [0025]. In the combination, client 201 includes firmware agent 125, so another device running a copy of client 201 would answer with its own firmware agent 125’s private key, and that signature would not verify with the stored certificate. *Id.*

298. A POSITA would have been motivated to make this check because, without it, system 200 would apply the first mobile device’s policy to a device running a copy of client 201. That device would be treated as the first mobile device and would receive that device’s security policies through the session, even though it is not the managed device to which system 200 issued the token. That result would undermine Wright’s safeguard “against unauthorized or rogue mobile devices trying to hijack the security policies or corporate data.” Wright, [0052]. Durham’s device-

bound identity and fresh nonce keep system 200 from relying on a copied credential or a replayed response. Durham, [0017], [0019], [0025], [0039].

299. A POSITA would have had a reasonable expectation of success because Durham already teaches verifying the firmware agent's signature, Wright already keeps per-device client management information in which the certificate would be stored, and Wright's session protocols are designed to accept new verbs. Durham, [0029]; Wright, [0104], [0133]. At the time of the check, system 200 would hold both inputs it needs: the stored certificate, retrieved with the device's client management information using the presented token, and the signed response returned in the current session. The added programming would apply Durham's signature verification to those inputs and act on the result.

300. Accordingly, Wright-Durham teaches that system 200 verifies firmware agent 125's signed response against the Identity Certificate stored with the token, which determines whether the device presenting the token (*authenticating end-user device*) is the first mobile device that was issued the token (*determining whether the authenticating end-user device is the first end-user device*). Durham, [0017], [0019], [0025], [0027]–[0029]; Wright, [0104], [0133].

- e. **8[e]: “enforcing the first access network control policy when the second device credential is the first device credential and the authenticating end-user device is the first end-user device; and”**

301. In my opinion, Wright-Durham teaches 8[e].

302. When the presented token is the first token (8[c]), the *authenticating end-user device* has passed the same authorization and authentication described for 1[d], and firmware agent 125’s response verifies against the stored Identity Certificate (8[d]), system 200 has confirmed that the device presenting the token is the first mobile device. System 200 then applies the first mobile device’s policy, which is the *first access network control policy* described for 8[a]. Wright, [0128], [0137]; Durham, [0029].

303. System 200 applies that policy by executing its first portion, as explained for claims 5 and 13 in Ground 1A. It proceeds to steps 328/327 of Wright’s Figure 3B and sends any updated security rules. Wright, [0128], [0146]–[0147]. On the device, authorization module 245 authenticates the downloaded policy, policy setting module 212 “receives, installs and updates” it and communicates it to policy enforcement control module 214, and module 214 enforces it through “one or more enforcement mechanism modules as specified by the policy.” *Id.*, [0061], [0068], [0132].

304. In my opinion, by sending the first portion's security rules for modules 212, 214, and 245 to authenticate, install, and enforce, system 200 directs client 201's agents to enforce the first mobile device's policy, and thereby enforces that policy. Wright, [0068], [0128], [0132], [0146]–[0147]. If no update is available, client 201 continues to enforce the security rules it already has, which are the first mobile device's rules. *Id.*, [0128].

305. In my opinion, once both checks succeed, firmware agent 125 also would no longer hold the first mobile device to Durham's default configuration. Durham applies that default only “until an acceptable security policy can be verified”; when a host's connection passes, Durham's network returns traffic rules that “may be applied with a high degree of assurance via firmware agent 125,” and the agent installs a verified policy by “updating network filters.” Durham, [0032], [0040], [0046]. In the combination, system 200 performs that verifying role: when the 1[d] authentication and the 8[d] check both succeed, system 200 would likewise direct firmware agent 125 to release the device from the default configuration, so that the device's traffic beyond network access control exchanges is governed by the first portion's security rules that modules 212, 214, and 245 enforce. Wright, [0068], [0128], [0132]; Durham, [0040].

306. Accordingly, system 200 applies the first mobile device's policy (*enforcing the first access network control policy*) when the presented token is the

first token (*when the second device credential is the first device credential*) and the firmware agent's response verifies against the stored Identity Certificate (*and the authenticating end-user device is the first end-user device*). Wright, [0068], [0128], [0132], [0146]–[0147]; Durham, [0025], [0029].

f. 8[f]: “enforcing a second access network control policy when the second device credential is not the first device credential or the authenticating end-user device is not the first end-user device.”

307. In my opinion, Wright-Durham teaches 8[f].

308. 8[f] is disjunctive. As explained in the Claim Construction section, I understand that a disjunctive limitation is met by one of its alternatives. My analysis addresses the second alternative: *the authenticating end-user device is not the first end-user device*.

309. A device may present the first token and pass the same authorization and authentication described for 1[d], yet its firmware agent's response may fail to verify against the stored Identity Certificate. As explained for 8[c] and 8[d], that is the case of a device running a copy of client 201, and the failed verification shows that the device is not the first mobile device. Wright, [0128]–[0130], [0137]; Durham, [0019], [0025], [0029].

310. In my opinion, a POSITA would have been motivated to have system 200 restrict that device's network access. The device has shown that it holds the first

mobile device's token and the enterprise's key, but not that it is the device to which system 200 issued the token, so it is the kind of "unauthorized or rogue mobile device[]" against which Wright's authorization module 232 is a safeguard. Wright, [0052]. Restricting its access is consistent with that safeguard and with the reasons for the combination explained in the motivation-to-combine section of this ground.

311. Durham does not describe denying access because a device is not the device that was issued a credential. In my opinion, however, Durham suggests doing so and provides the mechanisms for it. Durham restricts a host's access when circumstances indicate that the host may be circumventing restrictions, and it provides both a way for the network to send the host's firmware agent instructions when access is denied and a restrictive configuration that the agent enforces. Durham, [0021], [0030]–[0031], [0039], [0040], [0046]. I describe those teachings next.

312. First, Durham restricts a host's access when circumstances indicate that the host may be circumventing restrictions. Firmware agent 125 can "detect circumvention of the physical network interfaces under control of firmware agent 125," for example, when a user plugs in a Universal Serial Bus network interface over which the agent has no control, and can "report this violation to network access PDP 150 so the host system may be provided restricted network access." Durham, [0039]. Second, when access is denied, the network may return "instructions for

remediation.” *Id.*, [0031]. Those instructions reach firmware agent 125: PDP 150 “may use the secure channel to return signed, and optionally encrypted, result, to firmware agent 125,” whose attribute-value pairs may carry “instructions on how to fix or remediate host platform 100.” *Id.*, [0030]. Likewise, “[t]he host platform may receive a policy through a control channel that is passed to the firmware agent.” *Id.*, [0046]. Third, under Durham’s default configuration, “filters 120 block all access by host system 100” except network access control exchanges, which restricts a host “until an acceptable security policy can be verified.” *Id.*, [0021], [0040].

313. In my opinion, when the certificate check fails, a POSITA would have been motivated to have system 200 send remediation instructions over the established session and direct firmware agent 125 to apply Durham’s default configuration, keeping the device restricted until the discrepancy is resolved. The failed check is a circumstance like the circumvention Durham describes: it indicates that the device is attempting to obtain access to which it is not entitled, here by presenting another device’s credentials rather than by using an uncontrolled network interface, and Durham’s response to such a circumstance is restricted network access. Durham, [0039]. Durham’s remediation instructions give the network a way to tell the device what must be done, and its default configuration gives firmware agent 125 a restrictive state that keeps the device from reaching network resources while still allowing the network access control exchanges through which the device

can later be verified. *Id.*, [0021], [0030]–[0031], [0040], [0046]. Because the device passed 1[d], its session with system 200 is already established, so system 200 would send its instructions through that session. Wright, [0133], [0137].

314. Because system 200 detects the failed check and directs firmware agent 125 to apply the default configuration, the applied default configuration is a *second access network control policy* enforced by system 200. It is a policy: Durham itself calls it a “predefined default policy,” a defined set of filter rules that determines what traffic the device may send and receive. Durham, [0040]; *see id.*, [0022]–[0023]. It controls access to the network because those rules restrict the device to network access control exchanges, and it is a second policy because it differs from the first mobile device’s policy, which system 200 applies only when both checks succeed, as explained for 8[e]. *Id.*, [0021], [0040]. System 200 enforces it in the same way it enforces the first policy, by directing the device’s agent to apply it.

315. A POSITA would have had a reasonable expectation of success because Durham describes both pieces: the network returning remediation instructions to firmware agent 125 in a signed result over the secure channel when access is denied, and firmware agent 125 applying its default configuration. Durham, [0021], [0030]–[0031], [0046]. Durham’s firmware agent already applies a default configuration after booting, on a network link change, and when a received policy fails verification. *Id.*, [0021], [0040], [0046]. The added programming in system 200

would send the instructions when the 8[d] check fails, using the session through which system 200 already communicates with client 201. Wright, [0133].

316. Accordingly, Wright-Durham teaches enforcing the default configuration (*enforcing a second access network control policy*) when a device presents the same token as the first mobile device but is determined, in 8[d], not to be the first mobile device (*when ... the authenticating end-user device is not the first end-user device*). Durham, [0021], [0030]–[0031], [0039], [0040], [0046]; Wright, [0052].

4. **Claim 9: “The method recited in claim 8, wherein the second access network control policy differs from the first access network control policy in a network access privilege, an allowable network destination, a service usage accounting or billing policy, a service usage rate, a service speed or quality, an allowed access network, a user notification, a roaming policy or permission, an operation of the end-user device, or a combination of these.”**

317. In my opinion, Wright-Durham teaches claim 9.

318. Claim 9 depends from claim 8. The *second access network control policy* is Durham’s default configuration, which permits only network access control exchanges, as explained for 8[f]. Durham, [0015], [0021]. The *first access network control policy* is the 1[b] policy, which system 200 enforces under 8[e] through its first portion: the security rules that govern the device’s network use and restrict particular network services or applications, as explained for 8[a] and 8[e] and for

claim 11 in Ground 1A. Wright, [0014], [0048], [0270]. For example, a Wright security policy “may not allow a particular network service or application or both to be used,” and Wright’s enterprise policies address “program usage, network access, hardware restrictions, VPN access, data access, and many other policies.” *Id.*, [0014], [0270].

319. In my opinion, the two policies differ in what network access they permit. The *second access network control policy* allows only control exchanges with the network: under it, firmware agent 125 “may prevent any other traffic from entering or exiting host platform 100 other than the network access control channel.” Durham, [0015], [0021]. The first policy permits network use that is not limited to control exchanges, as explained for 8[e]: the device may use the network services and applications that its security rules allow, subject to the restrictions those rules impose. Wright, [0014], [0048], [0270]. That is a difference in a *network access privilege*: the first policy grants the device the privilege of using the network beyond control exchanges, and the second withholds it.

320. That the first policy’s second portion drops the connection when authentication fails does not change this conclusion. Wright, [0137]. The second portion applies only when the device fails authentication; under 8[e], the device has passed authentication and the certificate check, and system 200 applies the first

policy through its first portion. Only the first policy grants access beyond control exchanges.

321. Accordingly, Wright-Durham teaches that Durham's default configuration (*the second access network control policy*) differs from the 1[b] policy (*differs from the first access network control policy*) because the default configuration permits only control exchanges while the 1[b] policy permits network use beyond control exchanges (*in a network access privilege*). Durham, [0015], [0021]; Wright, [0014], [0048], [0137], [0270].

5. Claim 12: “The method recited in claim 8, wherein enforcing the second access network control policy comprises suspending or blocking network access by the authenticating end-user device.”

322. In my opinion, Wright-Durham teaches claim 12.

323. Claim 12 depends from claim 8 and recites that enforcing the second policy comprises “suspending or blocking” network access; the combination performs the blocking alternative. When the *authenticating end-user device* fails the certificate check, system 200 would direct firmware agent 125 to apply its default configuration, which is the *second access network control policy*, as explained for 8[f]. Under that default, “filters 120 block all access by host system 100” except network access control exchanges, so firmware agent 125 “may prevent any other

traffic from entering or exiting host platform 100 other than the network access control channel.” Durham, [0021], [0015].

324. A POSITA would have understood that this default lets the device only negotiate with the network, not reach network resources. The network access control channel carries the exchanges by which the device seeks access and receives instructions, such as Wright’s status request and session and Durham’s remediation instructions; it does not carry the device’s ordinary traffic to network services. Durham, [0015], [0021], [0031], [0046]. With all other traffic stopped at filters 120, the device’s network access is blocked (*blocking network access by the authenticating end-user device*).

325. Accordingly, Wright-Durham teaches that system 200 enforces Durham’s default configuration (*enforcing the second access network control policy*) through firmware agent 125’s filters 120, which stop all of the authenticating device’s traffic other than network access control exchanges (*comprises ... blocking network access by the authenticating end-user device*). Durham, [0015], [0021].

6. Claim 23

- a. **23[a]: “The method recited in claim 1, further comprising: detecting a service usage indication associated with the end-user device;”**

326. In my opinion, Wright teaches 23[a].

327. Claim 23 depends from claim 1. As explained for 1[a] in Ground 1A, Wright’s client mobile devices each send “a status request upon network access starting up.” Wright, [0128]. On the device, policy setting module 212 determines whether a network connection has been activated, and, “responsive to a network connection being activated,” authorization module 245 sends “a status request checking if new security information is available along with authorization information to the server system 200.” *Id.*, [0129], Fig. 3C. System 200 receives that request before it determines whether the exchange is authorized and authenticated. *Id.*, [0128], [0133]–[0138].

328. In my opinion, by receiving the status request, system 200 *detect[s] a service usage indication associated with the end-user device*. The status request tells system 200 that a particular mobile device has activated a network connection and is using it to communicate with system 200. Wright, [0128]–[0129]. The request is associated with that device because it carries the device’s authorization information, which identifies the device to which system 200 issued it. *Id.*, [0078], [0128].

329. A POSITA would have understood the status request to be an indication of service usage because it is triggered by activation of the device’s network connection and reflects the device’s use of that connection to communicate with system 200. Wright, [0128]–[0129]. The device sends the request only “responsive

to a network connection being activated,” and it sends the request over that connection. *Id.*, [0129].

330. Accordingly, Wright teaches that, by receiving the status request that client 201 sends when the device’s network connection is activated, system 200 *detect[s] a service usage indication associated with the end-user device*. Wright, [0128]–[0129].

b. 23[b]: “detecting a specified condition; and”

331. In my opinion, Wright teaches 23[b].

332. After receiving the status request, system 200 obtains client 201’s SHA1 response, as explained for 1[c][i] and 1[d] in Ground 1A. System 200 sends a random challenge, and client 201 computes and returns “SHA1(random challenge, key).” Wright, [0134]–[0136]. The key is the enterprise’s key, which Wright uses for “authentication of clients”; “[c]lients installed by some other enterprise will have a different key, and clients installed in an unmanaged mode will not have a key.” *Id.*, [0130]. System 200 then “computes the same SHA1 hash” and determines whether it “doesn’t match what the client sent.” *Id.*, [0137].

333. A mismatch (*detecting a specified condition*) means that client 201 has not shown that it holds the enterprise’s key, as a managed client must. Wright, [0130], [0135]–[0137]. The mismatch is a defined result of Wright’s authentication check, on which system 200 “drops the connection with no further ado.” *Id.*, [0137].

334. Accordingly, Wright teaches that system 200 determines that client 201's SHA1 response does not match the hash that system 200 computed (*detecting a specified condition*). Wright, [0130], [0135]–[0137].

c. 23[c]: “declaring a service processor integrity violation based on detecting the specified condition.”

335. In my opinion, Wright-Durham teaches 23[c].

336. Wright detects failed authentication and provides administrator feedback about security conditions. On the client, “[r]esponsive to a session not being authorized or being established,” client diagnostics module 246 “creates 328 an audit event noting failure to establish the session.” Wright, [0138]. On system 200, monitoring module 226 “receives diagnostic information such as events or audit logs from a client device and stores the information in a data object (242) for the client device.” *Id.*, [0054]. Wright’s policies also provide “feedback (e.g. custom error messages) to alert an administrator via a user interface screen using the server system 200 of certain conditions.” *Id.*, [0048]. And Wright teaches relaying to a policy server a report that protective services “have been disabled or removed from the system,” because that information “indicates a possible compromise of the protective system.” *Id.*, [0255]. The components that make and relay that report include implementation examples of client 201’s modules, and diagnostics module

844, which relays it, performs functions similar to those of client diagnostics module 246, providing status, error logs, and audit logs to the server. *Id.*, [0268]–[0269].

337. Durham likewise treats modified or withheld integrity measurements as a condition that “can be detected at the server” and that warrants “appropriate remedial action.” Durham, [0026]. And when an attack on the network access control communication is detected, “network administrators may be informed.” *Id.*, [0032].

338. In my opinion, an SHA1 mismatch is an analogous failure. Because the enterprise’s managed clients hold the enterprise’s key, a client that answers the challenge incorrectly is not operating as a managed client should: the mismatch indicates that client 201 may be compromised or misconfigured, or that the software answering for the device is not a managed client at all. Wright, [0130], [0137]. Each of these is a failure of the device-side component on which system 200 relies, just as Durham’s modified or withheld measurements are. Durham, [0026].

339. In my opinion, in view of Durham, a POSITA would have found it obvious for system 200 to record an SHA1 mismatch as a service processor integrity violation in the device’s client management information and to report it to the administrator (*declaring a service processor integrity violation based on detecting the specified condition*). Wright, [0048], [0104]; Durham, [0026], [0032]. System 200 would know which device’s record to update because the token in the status

request identifies the device's client management information, as explained for 1[b] in Ground 1A. Wright, [0104], [0128]. System 200 would issue the alert through the administrator feedback that Wright's policies already provide. *Id.*, [0048].

340. A POSITA would have been motivated to make that record and report because it distinguishes authenticated managed clients from clients that failed authentication, which serves Wright's "safeguard against unauthorized or rogue mobile devices." Wright, [0052]. Wright already treats a comparable client-side condition, protective services that have been disabled or removed, as indicating "a possible compromise of the protective system" to be relayed to the server, and Durham treats modified or withheld measurements as a condition to be detected at the server and acted upon. *Id.*, [0255]; Durham, [0026]. A record of the integrity violation lets the administrator identify the device involved and respond.

341. A POSITA would have had a reasonable expectation of success because system 200's modules already communicate with one another, keep per-device diagnostic information, and provide administrator feedback. Wright states that each module of system 200 "has a communication interface or is communicatively coupled to each of the other modules and has access to data objects 242," monitoring module 226 stores a client's diagnostic information in a data object for that client device, diagnosis module 228 analyzes that stored information, and Wright's policies provide feedback to alert an administrator. Wright, [0047]–[0048], [0054]–

[0055], [0104]. The added programming would only use the existing comparison result to write the record and issue the alert.

342. Accordingly, Wright-Durham teaches that system 200 records an SHA1 mismatch as a service processor integrity violation in the device's client management information and reports it to the administrator (*declaring a service processor integrity violation based on detecting the specified condition*). Wright, [0048], [0052], [0054], [0104], [0130], [0137]; Durham, [0026], [0032].

7. **Claim 24: “The method recited in claim 23, wherein the specified condition comprises that the service processor does not communicate with the network system within a predetermined amount of time after the network system detects the service usage indication.”**

343. In my opinion, Wright-Durham teaches claim 24.

344. Claim 24 depends from claim 23. Client 201 sends a status request “upon network access starting up,” which system 200 detects as the *service usage indication* of 23[a]. Wright, [0128]. The rest of client 201's communication in the exchange follows that request: the “Establish Session” protocol, in which client 201 sends its protocol version and user identification and then its SHA1 response to system 200's random challenge, and, in the combination, firmware agent 125's signed response, as explained in the motivation-to-combine section of this ground and for 8[d]. *Id.*, [0128], [0133]–[0137].

345. Durham teaches detecting at the server when an agent does not send expected integrity measurements: “if a rogue trust agent modifies or does not send firmware agent 125 integrity measurements, this action can be detected at the server.” Durham, [0026]. Durham also teaches that proof-of-liveness between firmware agent 125 and the PDP “may defeat replay and delayed messages attacks on the host platform.” *Id.*, [0039].

346. In my opinion, rejecting delayed messages requires a time limit. A nonce lets a receiver reject a response that carries an earlier exchange’s nonce, which is how it defeats replay. But a response that an attacker holds back and releases later in the same exchange still carries the current nonce, and the nonce alone cannot reveal the delay. Durham lists “delayed messages attacks” separately from replay attacks. Durham, [0039]. A message is delayed only relative to a time by which it was expected, so to defeat a delayed-message attack, the receiver must decline a response that arrives after some bound, for example, by honoring a nonce only for a limited time after issuing it. A POSITA therefore would have understood Durham’s proof-of-liveness to impose a time limit on the response it expects. *Id.*, [0027], [0039]. Likewise, a server can detect that an agent “does not send” expected measurements only by deciding, at some point, that the measurements are not coming. *Id.*, [0026].

347. A device that starts a connection but does not complete the exchange, or completes it late, gives system 200 no assurance that a live, managed client is present. Its status request shows that it is using the network, but without the communications that follow, system 200 has neither authenticated client 201 nor confirmed the device's identity. Wright, [0128], [0133]–[0137]. Leaving the device in that unresolved state would undermine Wright's safeguard "against unauthorized or rogue mobile devices." Wright, [0052].

348. In my opinion, a POSITA therefore would have found it obvious for system 200 to set a predetermined deadline when it receives the status request and to declare the integrity violation described for 23[c] if client 201 has not sent its next required communication by that deadline. System 200 knows which communication it expects next, whether the first "Establish Session" message (protocol version and user identification), the SHA1 response to its random challenge, or firmware agent 125's signed response, because system 200 conducts that exchange. Wright, [0128], [0133]–[0137]. When the status request arrives, system 200 would record the time and set the deadline (*a predetermined amount of time*); if client 201 does not send its next required communication by then (*the service processor does not communicate with the network system*), system 200 would thereby detect the *specified condition* of 23[b], as claim 24 defines it, and would record the integrity violation and alert the administrator, as explained for 23[c].

349. A POSITA would have had a reasonable expectation of success because system 200 already tracks each device's "Connection state," including "Last connected time," and receives client communications, including diagnostic information sent "upon an initial network connection by the mobile device with the server." Wright, [0054], [0107], [0109]. Recording the arrival time of a status request and comparing the current time with a deadline uses the same kind of time information that Wright already keeps. *Id.*, [0109]. The added programming would only start a timer on the status request and check for a timely response.

350. Accordingly, Wright-Durham teaches that the condition on which system 200 declares the integrity violation (*the specified condition comprises*) is that client 201 (*that the service processor*) has not sent its next required communication to system 200 (*does not communicate with the network system*) by the deadline (*within a predetermined amount of time*) that system 200 set when it received the status request (*after the network system detects the service usage indication*). Wright, [0052], [0054], [0107], [0109], [0128], [0133]–[0137]; Durham, [0026], [0039].

8. Claim 25: "The method recited in claim 23, wherein the specified condition comprises that the network system is unable to complete the end-user device service processor authentication procedure."

351. In my opinion, Wright-Durham teaches claim 25.

352. Claim 25 depends from claim 23. The *specified condition* is the SHA1 hash mismatch described for 23[b]. On a mismatch, Wright’s server “drops the connection with no further ado”; otherwise, “after this protocol is finished, the server has authenticated the client.” Wright, [0137].

353. In my opinion, a mismatch therefore leaves system 200 unable to complete the *end-user device service processor authentication procedure* of 1[d]. That procedure is the “Establish Session” protocol, at the end of which system 200 has authenticated client 201 only if the response matches. Wright, [0133]–[0137]. On a mismatch, system 200 drops the connection before that point, so the procedure ends without authenticating client 201. *Id.*, [0137]. As explained for 23[c], that mismatch is the condition on which system 200 would declare the service processor integrity violation.

354. Accordingly, Wright-Durham teaches that the SHA1 mismatch on which system 200 declares the integrity violation leaves system 200 unable to complete its Establish Session authentication of client 201 (*wherein the specified condition comprises that the network system is unable to complete the end-user device service processor authentication procedure*). Wright, [0133]–[0137]; Durham, [0026].

D. GROUND 1D: CLAIMS 16, 18, 23, 26, 27, 30, AND 33 ARE OBVIOUS OVER WRIGHT, BITTMANN, AND DURHAM

1. Motivation to Combine Wright, Bittmann, and Durham

355. In my opinion, a POSITA would have found it obvious to combine Wright, Bittmann, and Durham in the manner described in this section. I described Bittmann and Durham in the Overview of Bittmann and Overview of Durham sections above. The combination builds on the Wright-Bittmann combination explained in Ground 1B and the Wright-Durham combination explained in Ground 1C, for the reasons given in the motivation-to-combine sections of those grounds. I explain below how the combined system would be arranged, why a POSITA would have been motivated to arrange it that way, and why a POSITA would have had a reasonable expectation of success.

356. Ground 1D keeps the Wright-Bittmann arrangement explained in the motivation-to-combine section of Ground 1B. In that arrangement, client 201 includes Bittmann's monitoring unit 130, whose report generator 420 reports the mobile device's usage to system 200, and system 200 includes Bittmann's server utility 150, in which mediation system 210 receives and correlates the reports, rating system 220 rates the reported activity, and allowance manager 230 updates the allowances of the account associated with the device. Bittmann, [0042]–[0043], [0054]. Because those reports travel in the sessions that Wright's "Establish Session" protocol authenticates, system 200 continues to perform the claim 1 operations

explained in Ground 1A: it receives the device's encrypted token, obtains the policy applicable to the device, and authenticates client 201 by comparing client 201's SHA1 response with the hash that system 200 computes itself. Wright, [0078], [0104], [0128]–[0129], [0133]–[0137].

357. Ground 1D also adds Durham's firmware agent 125 to client 201, as in the Wright-Durham combination explained in the motivation-to-combine section of Ground 1C. Client 201 retains Wright's policy setting module 212, policy enforcement control module 214, and authorization module 245, the device agents explained for 1[c][ii] in Ground 1A, which are software executing on the mobile device, together with monitoring unit 130, which in Wright-Bittmann is implemented, as Bittmann permits, as part of that software. Wright, [0046], [0058]–[0061], [0068]; Bittmann, [0053]. Firmware agent 125 operates alongside that software on the same device, in the protected environment that Durham describes. Durham, [0012]–[0013], [0048]. As in Wright-Durham, system 200 performs the network-side functions of Durham's PDP 150 and trust server 155: it issues a nonce and verifies firmware agent 125's signed responses using the public key in the agent's Identity Certificate, which Durham establishes "using Public-Key/Private-Key Identity Certificates" and which system 200 stores with the device's token in the device's client management record, as explained for 8[b] and 8[d] in Ground 1C. Durham, [0017], [0019], [0025], [0027]–[0029]; Wright, [0104].

358. Bittmann describes the monitor that firmware agent 125 would implement. Monitoring unit 130 includes monitor 410 “for monitoring data activity of the associated communication device” and report generator 420 “for generating reports on the data activity.” Bittmann, [0054]. Bittmann does not limit the monitor to software: monitoring unit 130 “may be any combination of hardware, software and/or firmware that performs the functions as defined and explained herein.” *Id.*, [0053]. Bittmann places the monitor where the device’s data passes, linking it between the device’s CPU 330 and network interface 310 or to bus 350 connecting them. *Id.*, [0052]–[0053], Figs. 3A–3B. Its reports can identify, among other things, the “volume of data transferred (e.g. kilobytes, number of packets, etc.),” the “time-interval for data transfer,” and the “source and destination of data.” *Id.*, [0060].

359. In the combination, firmware agent 125 would implement a second, independent instance of Bittmann’s monitor for the device’s network traffic. Bittmann, [0053]. Durham explains that firmware agent 125 “may serve as a tamper-resistant execution environment on host platform 100” and “may host code that attests to the integrity of the software on host platform 100.” Durham, [0013]. Durham’s Figure 3 shows that the embedded firmware agent has its own components on Durham’s agent bus 300: Durham’s dynamic memory, non-volatile storage, and control circuitry 330. *Id.*, [0047]–[0049], Fig. 3. Durham’s control circuitry 330 executes instructions from that memory and storage, so the agent has “sufficient

functionality to perform operations independent of the host platform and/or host operating system.” Id., [0049]. A POSITA would have understood that an agent that hosts and executes its own code in this way is able to host the code of Bittmann’s monitor as well. Firmware agent 125 also “has exclusive access to host resources including, for example, traffic filters 120 on a network interface card,” which it configures through “an exclusive internal bus,” and through those filters it controls the traffic “entering or exiting host platform 100.” Id., [0012], [0015], [0021]. Because the device’s network traffic enters and exits through those filters, and firmware agent 125 alone configures and controls them, a POSITA would have understood that the agent, which determines at the filters whether each packet passes, is positioned to count the packets and bytes that pass. That is the same counting that Bittmann’s monitor performs where monitoring unit 130 “links between” Bittmann’s CPU 330 and network interface 310. Bittmann, [0052], Fig. 3A. Although Durham’s traffic filters may be hardware or software, in the combination the agent counts at traffic filters 120 on the network interface card, not at software filters in the host. Durham, [0012], [0021]. Firmware agent 125 would therefore host the functions of monitor 410 and report generator 420, count the device’s network traffic as that traffic crosses the network interface whose filters firmware agent 125 controls, and keep those counts in its own memory. That second monitor would

measure the same traffic that monitoring unit 130 measures in client 201's software, but it would do so independently of that software.

360. System 200 would thus receive two reports of the device's usage. Monitoring unit 130 in client 201's software would continue to send its report, as in Wright-Bittmann. Bittmann, [0054], [0062]. Firmware agent 125 would send its own report of the traffic it counted through the authenticated reporting path that Durham describes. Durham's signed exchange occurs when the host seeks network access, but the device's usage accrues afterward. Durham, [0027]. System 200 would therefore request the agent's report in an exchange that it adds to the session as a new verb, with a fresh nonce, just as it adds Durham's exchange as a new verb for 8[d] in Ground 1C, and it would do so periodically, as Bittmann's server utility 150 initiates communication sessions to obtain device reports. Wright, [0133]; Bittmann, [0062]. In that exchange, firmware agent 125 would include its counts among the attribute-value pairs that it signs with its private key, together with the nonce that system 200 issued for the exchange. Durham, [0025], [0027]–[0028]. That signed report is firmware agent 125's trusted report, for the reasons I explain for 16[b]. Both reports describe the same device's use of the network during the same token-opened session.

361. A POSITA would have been motivated to arrange the combination this way for the reasons explained in the motivation-to-combine section of Ground 1B:

to have system 200 check the device's own usage report against an independent report that client 201 cannot alter. Wright's authorization module 232 authorizes client 201's communication exchange, and Establish Session authenticates client 201 when a matching SHA1 response shows that client 201 held the enterprise's key when the session began. Wright, [0128], [0130], [0137]. Neither check shows that client 201 was not compromised afterward, including in how it reports its activity. A client whose software is tampered with after it authenticates remains in its authenticated session, and its report can understate or omit activity. A discrepancy between the device's report and an independent report of the same activity would reveal that the client is misreporting its activity.

362. A POSITA would have recognized firmware agent 125 as a suitable independent source for that report because the device's host software cannot tamper with it. Durham explains that "no code executing within host operating system 110 can tamper with firmware agent code, prevent firmware agent code from running, or circumvent operation of the firmware agent." Durham, [0012]. With those properties, firmware agent 125 "may serve as a tamper-resistant execution environment on host platform 100." *Id.*, [0013]. Firmware agent 125 also has exclusive access to the traffic filters on the network interface card, which it configures through "an exclusive internal bus." *Id.*, [0012], [0021]. Durham uses those properties so that firmware agent 125 can serve as "a trusted policy

enforcement point (PEP) on the host platform to enforce policies even when the host operating system is compromised.” *Id.*, [0010]. A monitor hosted in firmware agent 125 would therefore produce a usage report that remains reliable when client 201’s software does not.

363. Bittmann itself contemplates protecting device reporting from tampering: the device or monitoring unit 130 may include “safety features intended to block the ability to tamper with the device or with the automatic transmission feature, so as to ensure integrity of the activity data transmitted to server utility 150.” Bittmann, [0057]. Durham’s firmware agent 125 provides a concrete architecture for implementing such a tamper-protected monitor. In my opinion, a POSITA would have been motivated to keep monitoring unit 130 in client 201’s software alongside the trusted monitor in firmware agent 125, because having both reports allows system 200 to compare them. Monitoring unit 130 runs in host software without Durham’s protection, while the monitor in firmware agent 125 counts the same traffic where host software cannot reach it. Durham, [0012]–[0013], [0021]. The firmware report alone would give system 200 a reliable measure of the device’s usage, but it would not show whether client 201’s software is reporting accurately. A comparison of the two reports would show it, because a discrepancy between them would expose misreporting by client 201’s software, which is the same software that enforces the device’s policy.

364. Comparing the reports would also serve Wright's stated purpose. Wright's authorization module 232 is "a further safeguard against unauthorized or rogue mobile devices trying to hijack the security policies or corporate data." Wright, [0052]. A client that is compromised after it authenticates, and that understates its activity while it retrieves corporate data, would not be caught by Wright's authentication, but its understated report would not match the trusted monitor's count of the same traffic. Comparing the reports would therefore extend Wright's safeguard against rogue mobile devices to clients that are compromised after they authenticate.

365. Claims 23, 26, 27, and 30 also require declaring a service processor integrity violation. In my opinion, a POSITA would have been motivated to treat either of two conditions as such a violation: a discrepancy between the two reports, or either report showing restricted activity continuing after the allowance reached its restriction level and system 200 directed the restriction. Either condition indicates that the device is not behaving as a managed client would be expected to behave. A managed client 201 reports its usage accurately, so its report agrees with the trusted monitor's count of the same traffic. And a managed client 201 applies the restriction that system 200 directs when an allowance reaches its restriction level, so the device's restricted activity stops. Bittmann, [0045], [0055], [0070]–[0071]. This motivation parallels the one explained for 23[c] in Ground 1C for the SHA1

mismatch. Wright contemplates acting on a suspected compromise: its components report whether other protective services “have been disabled or removed from the system,” and “[t]his information may be relayed to the policy server 850 as it indicates a possible compromise of the protective system.” Wright, [0255]. Durham likewise teaches that when firmware agent 125’s integrity measurements are modified or not sent, “this action can be detected at the server ... and appropriate remedial action can be taken.” Durham, [0026].

366. In my opinion, a POSITA would have had a reasonable expectation of success in incorporating Bittmann’s and Durham’s functionality into Wright and implementing the trusted and untrusted reports, the report reconciliation, and the integrity-violation functions, for the reasons explained in the motivation-to-combine sections of Grounds 1B and 1C and because the references already provide each part. Bittmann provides the monitor, its report generator, the usage fields its reports carry, and a mediation system that correlates and aggregates “reports from multiple sources to produce a unified report record,” and it permits the monitor to be implemented in firmware. Bittmann, [0042], [0053]–[0054], [0060]. Durham provides the tamper-resistant execution environment, whose control circuitry executes instructions from the agent’s own memory; firmware agent 125’s exclusive access to the traffic filters on the network interface; and an authenticated reporting path that the network verifies. Durham, [0012]–[0013], [0021], [0025], [0027]–[0029], [0049]. Wright

provides the per-device client management record, the storage of diagnostic information for each client device, and administrator feedback. Wright, [0048], [0054], [0104].

367. Implementing the combination would have required only following those teachings and routine engineering: loading the functions of Bittmann's monitor 410 and report generator 420 into firmware agent 125 at the network interface it controls; having that monitor keep its counts in firmware agent 125's memory; adding a session verb by which system 200 requests those counts with a fresh nonce, and having firmware agent 125 return them in the signed attribute-value pairs it already produces; programming mediation system 210, which already correlates reports from multiple sources, to match the trusted report with monitoring unit 130's report for the same session and interval and compare their counts; and having system 200 record the result of that comparison, or of the allowance check, in the device's client management record as explained for 23[c]. Bittmann, [0042], [0045], [0053]–[0054]; Durham, [0013], [0025], [0049]; Wright, [0048], [0104], [0133]. Each step uses a function that Wright, Bittmann, or Durham already describes.

2. Claims 16/18/33

a. Claim 16

- (i) 16[a]: “The method recited in claim 1, further comprising: receiving a device data usage report from the end-user device;”**

368. In my opinion, Wright-Bittmann-Durham teaches 16[a], for the same reasons I explained for 16[a] in Ground 1B.

369. Ground 1D keeps monitoring unit 130 in client 201 and server utility 150 in system 200, as explained in the motivation-to-combine section of this ground, and adding firmware agent 125 does not change monitoring unit 130 or its report. Monitor 410 monitors the device’s data activity, and report generator 420 generates reports on that activity for server utility 150, whose mediation system 210 receives them within system 200. Bittmann, [0042], [0054]. The report can identify, for example, the “volume of data transferred (e.g. kilobytes, number of packets, etc),” the “time-interval for data transfer,” and the “source and destination of data.” *Id.*, [0060]. System 200 thus receives monitoring unit 130’s report (*receiving a device data usage report*) from client 201 on the mobile device (*from the end-user device*).

370. Accordingly, Wright-Bittmann-Durham teaches that system 200 receives monitoring unit 130’s report (*receiving a device data usage report from the end-user device*). Bittmann, [0054], [0060].

(ii) 16[b]: “obtaining a trusted data usage report associated with the end-user device; and”

371. In my opinion, Wright-Bittmann-Durham teaches 16[b].

372. In the combination, firmware agent 125 in client 201 implements Bittmann’s monitor and reports the device’s network traffic to system 200, as explained in the motivation-to-combine section of this ground. The rest of client 201’s software, including monitoring unit 130, is analogous to Durham’s host software. Durham distinguishes firmware agent 125 from the host software on host platform 100, such as trust agent 105 and software agents 102, and it recognizes that such host software can be rogue: “a rogue trust agent” can modify or withhold firmware agent 125’s measurements. Durham, [0012], [0015], [0024], [0026]. Client 201’s modules are software executing on the mobile device, and in Wright-Bittmann monitoring unit 130 is implemented as part of that software, one of the implementations Bittmann permits. Wright, [0046]; Bittmann, [0053]. They occupy the position of Durham’s host software, and firmware agent 125 occupies the same protected position on the mobile device that it occupies on Durham’s host platform 100.

373. In my opinion, firmware agent 125’s report is trusted because client 201’s host software cannot alter, forge, or replay it. As to generating the report, “no code executing within host operating system 110 can tamper with firmware agent

code, prevent firmware agent code from running, or circumvent operation of the firmware agent.” Durham, [0012]. The monitor’s code therefore runs, and its counts accumulate in the dynamic memory on firmware agent 125’s own agent bus 300, where host software cannot change them. *Id.*, [0012]–[0013], [0048]–[0049]. Firmware agent 125 also has “exclusive access” to traffic filters 120 on the network interface card, which it configures through “an exclusive internal bus.” *Id.*, [0012], [0021]. Host software therefore cannot reconfigure the point at which the agent counts the device’s traffic or alter what the agent counts there: the traffic the agent counts is the traffic that actually crosses the network interface.

374. As to transmitting the report, firmware agent 125 would sign its report as it signs its responses in Wright-Durham, as explained for 8[d] in Ground 1C. Durham establishes the agent’s identity “using Public-Key/Private-Key Identity Certificates,” and the agent signs the attribute-value pairs it sends “with a private/symmetric key of firmware agent 125.” Durham, [0019], [0025]. Host code cannot tamper with firmware agent 125, the agent’s identity “may be further protected using Trusted Platform Module (TPM) 130,” and the agent keeps its instructions and data in its own dynamic memory and non-volatile storage on its own agent bus 300. *Id.*, [0012], [0017], [0048]–[0049]. A POSITA would therefore have understood that host software cannot extract the agent’s private key, so only firmware agent 125 can produce a signature that verifies with the public key in the

agent's certificate, as explained for 8[d] in Ground 1C. *Id.*, [0019], [0025]. Durham explains that the signature “provides assurance to a PDP component that the data is submitted by an entity possessing the pertinent keying material” and protects the data from modification en route. *Id.*, [0025]. The signed data also includes nonces: the agent includes an unpredictable nonce “to prove liveness of the PDP, and ensure that an intermediary rogue trust agent cannot capture and replay the measurements,” and it includes in its signature the nonce that the network issues for the exchange. *Id.*, [0025], [0027]–[0028]. Because system 200 issues a new nonce for each exchange, host software cannot pass off an earlier signed report as a current one.

375. Monitoring unit 130's report, in contrast, comes from host software that a compromised client is able to alter. Monitoring unit 130 runs within client 201's software, outside firmware agent 125's protected environment, so code running on the device can change what monitoring unit 130 counts or what report generator 420 sends. Durham recognizes that host software can be rogue, which is why it detects at the server a rogue trust agent that “modifies or does not send firmware agent 125 integrity measurements.” Durham, [0026]. Wright likewise recognizes that protective software on the device can be “disabled or removed from the system.” Wright, [0255]. In the combination, the protection that Bittmann contemplates for the device's reporting comes from firmware agent 125, not from monitoring unit 130, which remains part of client 201's software. Bittmann, [0057].

376. The trusted report is *associated with the end-user device*. Firmware agent 125's report would arrive in the session that the device opens with its token, and the token identifies the device's client management record, as explained for 1[b] in Ground 1A. Wright, [0104], [0128], [0133]. As with 8[d] in Ground 1C, system 200 would verify the report's signature with the public key in the firmware agent's Identity Certificate, which system 200 stored with the token in that record, as explained for 8[b] in Ground 1C. Durham's trust server evaluates the information that firmware agent 125 sends "first for integrity and authenticity of the data, using digital signature verification." Durham, [0029]. A verified signature thus ties the report to the enrolled firmware agent 125 of the device whose record the token identifies, and the report measures that device's own network traffic.

377. Accordingly, Wright-Bittmann-Durham teaches that system 200 obtains firmware agent 125's report (*obtaining a trusted data usage report associated with the end-user device*), a report that client 201's host software cannot alter, forge, or replay and that system 200 associates with the device through the token-opened session and the Identity Certificate stored in the device's client management record. Durham, [0012], [0019], [0021], [0025], [0027]–[0029]; Wright, [0104].

(iii) 16[c]: “reconciling the device data usage report and the trusted data usage report.”

378. In my opinion, Wright-Bittmann-Durham teaches 16[c], for substantially the same reasons I explained for 16[c] in Ground 1B, with firmware agent 125’s report as the trusted data usage report.

379. As explained for 16[c] in Ground 1B, mediation system 210 correlates and aggregates “reports from multiple sources to produce a unified report record,” which rating system 220 then rates. Bittmann, [0042]–[0043]. In Ground 1D, the two reports that mediation system 210 reconciles are monitoring unit 130’s report (*the device data usage report*) and firmware agent 125’s report (*the trusted data usage report*). Mediation system 210 would match the two reports by the token-opened session whose traffic each covers and by the interval each covers, compare the usage each reports for the same traffic, such as data volume, and combine them into one unified record for rating, so that the same usage is not rated twice (*reconciling*). Bittmann, [0042]–[0043], [0060]. Because Bittmann’s device reports may be sent periodically or at off-peak periods rather than as each transfer occurs, mediation system 210 would align the reports by the interval each covers rather than by when each arrives. *Id.*, [0062].

380. A difference between the reports would show that monitoring unit 130 misreported the device’s usage. Firmware agent 125 counts the device’s traffic

where host software cannot alter the counts, as explained for 16[b], so when the two reports disagree, the host-side report is the one that departs from the device's actual usage. Mediation system 210 would compare like with like: the same measure, such as bytes, over the same interval. It would use quantities that both reports carry, because both monitors implement Bittmann's monitor, such as the "volume of data transferred" and the "time-interval for data transfer." Bittmann, [0060]. A difference in the compared quantities reflects activity that monitoring unit 130's report omitted or misstated.

381. Accordingly, Wright-Bittmann-Durham teaches that mediation system 210 matches, compares, and combines (*reconciling*) monitoring unit 130's report (*the device data usage report*) and firmware agent 125's report (*and the trusted data usage report*). Bittmann, [0042]–[0043], [0060].

b. Claim 18: "The method recited in claim 16, wherein obtaining the trusted data usage report comprises obtaining the trusted data usage report from a software or firmware agent in a secure execution environment on the end-user device."

382. In my opinion, Wright-Bittmann-Durham teaches claim 18.

383. As explained for 16[b], system 200 *obtain[s] the trusted data usage report from* firmware agent 125 (*a software or firmware agent*) within client 201 on Wright's mobile device (*on the end-user device*). Firmware agent 125 runs on host platform 100, which Durham explains "is intended to represent a broad range of

electronic devices that may be coupled with a network.” Durham, [0011]–[0012]. In the combination, that device is Wright’s mobile device.

384. Durham explains that firmware agent 125 “may serve as a tamper-resistant execution environment on host platform 100” and “may host code that attests to the integrity of the software on host platform 100.” Durham, [0013]. The embedded firmware agent executes instructions with its own control circuitry, from its own dynamic memory and non-volatile storage, and has “sufficient functionality to perform operations independent of the host platform and/or host operating system.” *Id.*, [0049]. A POSITA would have understood that an agent that hosts and executes its own code in this way is able to host other code as well. In the combination, the code that firmware agent 125 hosts includes the monitoring and reporting code of the second instance of Bittmann’s monitor, as explained in the motivation-to-combine section of this ground. That monitoring and reporting code therefore executes in firmware agent 125’s tamper-resistant execution environment.

385. That environment is secure because no host code can tamper with it or circumvent it: “no code executing within host operating system 110 can tamper with firmware agent code, prevent firmware agent code from running, or circumvent operation of the firmware agent.” Durham, [0012]. A POSITA would have understood an execution environment that the device’s other software cannot tamper with, stop, or circumvent to be a secure execution environment, and, as explained

for 16[b], that protection is what makes firmware agent 125's report trusted. Thus, firmware agent 125 is in a *secure execution environment*.

386. Accordingly, Wright-Bittmann-Durham teaches that system 200 obtains firmware agent 125's report (*obtaining the trusted data usage report*) from firmware agent 125 (*from a software or firmware agent*), whose code executes in Durham's tamper-resistant execution environment (*in a secure execution environment*) on Wright's mobile device (*on the end-user device*). Durham, [0011]–[0013], [0049].

c. Claim 33: “The method recited in claim 18, wherein the software or firmware agent is part of the service processor.”

387. In my opinion, Wright-Bittmann-Durham teaches claim 33.

388. In the combination, firmware agent 125 (*the software or firmware agent*) operates within client 201 as a firmware component, alongside client 201's software modules and in the protected environment that Durham describes, as explained in the motivation-to-combine section of this ground and for 16[b]. Client 201 is Wright's *service processor*, as explained for 1[c][i] in Ground 1A, so firmware agent 125 is *part of the service processor*. Durham itself calls firmware agent 125 “[a] service processor.” Durham, [0012]. In the combination, however, firmware agent 125 is not a separate service processor but part of client 201's *service processor*: it reports the device's usage to system 200 alongside monitoring unit 130

and operates with client 201's policy setting module 212, policy enforcement control module 214, and authorization module 245 on the same device.

389. Including firmware agent 125 within client 201 is consistent with Wright's description of its modules, which "can be implemented as software, hardware, firmware or any combination of the three." Wright, [0286]. As explained in the motivation-to-combine section of Ground 1C, a firmware component of client 201 is as much a part of client 201 as its software modules.

390. Accordingly, Wright-Bittmann-Durham teaches that firmware agent 125 (*the software or firmware agent*), which sends the trusted report, operates within client 201 (*is part of the service processor*). Wright, [0286]; Durham, [0012].

3. Claims 23/26/27/30

a. Claim 23

- (i) 23[a]: "The method recited in claim 1, further comprising: detecting a service usage indication associated with the end-user device"**

391. In my opinion, Wright-Bittmann-Durham teaches 23[a].

392. In the combination, system 200 obtains firmware agent 125's trusted report of the device's data usage, which is associated with the device, as explained for 16[b]. By receiving that report, system 200 *detect[s] a service usage indication associated with the end-user device*: the report tells system 200 that the identified device used the network, and how much.

393. The report directly reflects the device's use of the network. The monitor in firmware agent 125 counts the data that the device sends and receives across the network interface, and its report carries usage information of the kind Bittmann's reports carry, such as the "volume of data transferred (e.g. kilobytes, number of packets, etc)." Bittmann, [0053]–[0054], [0060]. Each such report therefore indicates the device's use of network service.

394. System 200 also receives monitoring unit 130's report, as explained for 16[a]. System 200 uses that report with the trusted report to detect the mismatch described for 23[b].

395. Accordingly, Wright-Bittmann-Durham teaches that system 200 receives firmware agent 125's trusted report of the device's network usage (*detecting a service usage indication*), which system 200 associates with the mobile device (*associated with the end-user device*). Bittmann, [0060]; Durham, [0025], [0029].

(ii) 23[b]: "detecting a specified condition; and"

396. In my opinion, Wright-Bittmann-Durham teaches 23[b].

397. System 200 includes mediation system 210, which reconciles monitoring unit 130's report with firmware agent 125's trusted report, as explained for 16[c]. Bittmann, [0042]–[0043], [0060]. A difference between the reports shows that monitoring unit 130 misreported the device's usage, for the reasons explained

for 16[c] in Ground 1B and in this ground. Because host software cannot alter firmware agent 125's counts, a difference in the quantities compared as explained for 16[c] identifies the host-side report as inaccurate. System 200 thereby detects the mismatch (*detecting a specified condition*).

398. Alternatively, the *specified condition* is that either report shows use inconsistent with the device's service policy. In Wright-Bittmann, that policy includes Bittmann's allowance for the account associated with the device, as explained in the motivation-to-combine section of Ground 1B; an allowance can represent "a quantity of past or future activity, for example volume in bytes." Bittmann, [0038]. When an allowance reaches its "pre-determined restriction level," allowance manager 230 "causes the restriction of activity in the account." *Id.*, [0045]. In Bittmann's Figure 5 example, server utility 150 then tells the device's monitoring units to restrict activity, "for example to only allow free traffic (e.g. toll free web sites) and traffic to the network operator," and control 430 applies the restriction. *Id.*, [0055], [0070]–[0071]. A report showing that restricted activity continued after the allowance reached that level, and after system 200 directed the restriction, shows that the restriction did not take effect (*detecting a specified condition*).

399. System 200 would identify that activity from the reports themselves. Bittmann's reports can identify the "time of day" and the "time-interval for data

transfer,” which place the activity after system 200 directed the restriction. They can also identify the “source and destination of data” and “any web sites accessed or only those which charge for access,” which separate data that the device sends to destinations the restriction blocks from the “free traffic (e.g. toll free web sites) and traffic to the network operator” that Bittmann continues to allow. Bittmann, [0060], [0071]. That restricted activity, continuing after system 200 has directed the restriction, is what I mean in this ground by usage continuing beyond the restriction level, or usage over the device’s service policy. Usage that passes the restriction level before the restriction is directed is not that condition: in Bittmann’s own example, the allowance is “reached (and even overstepped)” before the monitoring units are told to restrict activity, and Bittmann allows activity to continue for a time after the level is reached. *Id.*, [0046], [0071]. Nor is the free traffic that Bittmann continues to allow during the restriction.

400. Accordingly, Wright-Bittmann-Durham teaches that system 200 detects either a difference between monitoring unit 130’s report and firmware agent 125’s report or, alternatively, a report showing restricted activity continuing after the allowance reached its restriction level and system 200 directed the restriction (*detecting a specified condition*). Bittmann, [0042]–[0043], [0045], [0060], [0070]–[0071].

(iii) 23[c]: “declaring a service processor integrity violation based on detecting the specified condition.”

401. In my opinion, Wright-Bittmann-Durham teaches 23[c].

402. Wright-Bittmann-Durham teaches 23[c] for the reasons I explained for 23[c] in Ground 1C, with either of the two conditions of 23[b], reports that do not match or a report showing use inconsistent with the service policy, in place of the SHA1 mismatch. As explained there, system 200 stores diagnostic information for each client device in a data object for that device, and Wright’s policies provide “feedback (e.g. custom error messages) to alert an administrator via a user interface screen using the server system 200 of certain conditions.” Wright, [0048], [0054]. A POSITA would have found it obvious for system 200 to record the condition as a service processor integrity violation in the device’s client management information and to report it to the administrator. *Id.*, [0048], [0104]. A POSITA would have had a reasonable expectation of success for the reasons explained there: system 200’s modules already communicate with one another, keep per-device diagnostic information, and provide administrator feedback, and the added programming would use the result of mediation system 210’s comparison, or of the allowance check, to write the record and issue the alert. *Id.*, [0047]–[0048], [0054]–[0055], [0104].

403. These conditions are analogous to the condition that Durham treats as warranting action at the server. Durham explains that “if a rogue trust agent modifies

or does not send firmware agent 125 integrity measurements, this action can be detected at the server ... and appropriate remedial action can be taken.” Durham, [0026]. In Durham, host software interferes with what the network learns about the platform, and the server detects that interference “based, at least in part, on associated security credentials.” *Id.* In the combination, client 201’s host software either misstates the device’s usage or fails to apply the restriction it was directed to apply, and system 200 detects that failure by checking monitoring unit 130’s report against firmware agent 125’s signed report, or by checking whether either report shows restricted activity continuing after system 200 directed the restriction. In each case, the server detects that the device’s host-side software has not behaved as a managed client’s software should.

404. If the reports do not match, the difference shows that monitoring unit 130’s report was altered or incomplete. Client 201’s host software cannot alter firmware agent 125’s report, as explained for 16[b], so a discrepancy in the quantities compared as explained for 16[c] lies in the host-side report: monitoring unit 130 either misstated activity that occurred or omitted it. Monitoring unit 130 is part of client 201, the *service processor*, so an altered or incomplete report is a failure of the service processor’s own reporting.

405. In my opinion, if usage is inconsistent with a service policy that client 201 enforces, such usage continuing beyond the restriction level after system 200

has directed the restriction shows that client 201 did not enforce that policy. In the combination, client 201 enforces the allowance restriction: when the allowance reaches its restriction level, system 200 directs monitoring unit 130 to restrict activity, and control 430 within monitoring unit 130 blocks the restricted traffic. Bittmann, [0045], [0055], [0070]–[0071]. Control 430 acts where monitoring unit 130 sits, between Bittmann’s CPU 330 and network interface 310, and it blocks restricted outgoing data before that data reaches network interface 310. *Id.*, [0052], [0055]. A managed client 201 that applied that restriction would therefore not send the restricted traffic. Ground 1D keeps the Wright-Bittmann arrangement, in which control 430 is part of monitoring unit 130 in client 201’s software, and it adds firmware agent 125 as a second, independent monitor, as explained in the motivation-to-combine section of this ground. So if client 201 does not apply the restriction, the restricted data that the device sends crosses the network interface, where firmware agent 125 counts it. If either report shows that restricted activity continued after the allowance reached its restriction level and after system 200 directed the restriction, then clearly client 201 did not enforce the policy it was directed to enforce. System 200 identifies that activity from the report fields explained for 23[b].

406. Either condition indicates possible tampering with client 201. Wright treats an analogous condition the same way: its components report whether other

protective services “have been disabled or removed from the system,” and “[t]his information may be relayed to the policy server 850 as it indicates a possible compromise of the protective system.” Wright, [0255]. Client 201 protects Wright’s mobile device, and a client that misreports its usage or fails to enforce its assigned restriction is not functioning as that protective software should.

407. Accordingly, in my opinion, a POSITA would have found it obvious for system 200 to record either a mismatch between the two usage reports or usage over the device’s service policy, that is, restricted activity continuing after system 200 directed the restriction (either of which is a *specified condition*) as a service processor integrity violation in the device’s client management record (*declaring a service processor integrity violation based on detecting the specified condition*). Wright, [0048], [0104].

- b. Claim 26: “The method recited in claim 23, wherein the service usage indication is a first service usage indication, and further comprising receiving a second service usage indication from the service processor, and wherein the specified condition comprises that the second service usage indication and the first service usage indication are inconsistent.”**

408. In my opinion, Wright-Bittmann-Durham teaches claim 26.

409. The *service usage indication* of 23[a] is firmware agent 125’s trusted report, which is a *first service usage indication*, as explained for 23[a]. System 200 also receives monitoring unit 130’s report from client 201 (*receiving a second*

service usage indication from the service processor), as explained for 16[a] and 23[a]. Client 201 is Wright's *service processor*, as explained for 1[c][i] in Ground 1A, and monitoring unit 130 operates within it. Bittmann, [0054].

410. The *specified condition* is the first alternative explained for 23[b]: a difference between the two reports (*the specified condition comprises that the second service usage indication and the first service usage indication are inconsistent*). System 200 declares *a service processor integrity violation based on detecting the specified condition* as explained for 23[c].

411. Accordingly, Wright-Bittmann-Durham teaches that system 200 receives firmware agent 125's report (*the service usage indication is a first service usage indication*) and monitoring unit 130's report (*receiving a second service usage indication*) from client 201 (*from the service processor*), and that the condition system 200 detects (*the specified condition comprises*) is that monitoring unit 130's report (*that the second service usage indication*) and firmware agent 125's report (*and the first service usage indication*) differ (*are inconsistent*). Bittmann, [0042]–[0043], [0054], [0060]; Durham, [0025].

- c. **Claim 27: “The method recited in claim 23, wherein the service usage indication is a first service usage indication, and further comprising receiving a second service usage indication from the service processor, and wherein the specified condition comprises that either the first service usage indication or the second service usage indication is inconsistent with the service policy.”**

412. In my opinion, Wright-Bittmann-Durham teaches claim 27.

413. As for claim 26, the *service usage indication* is firmware agent 125’s trusted report (*a first service usage indication*), and system 200 also receives monitoring unit 130’s report from client 201 (*receiving a second service usage indication from the service processor*), as explained for 23[a] and for 1[c][i] in Ground 1A.

414. The device’s *service policy* is the device-level policy that system 200 obtains from the device’s client management record, as explained for 1[b] in Ground 1A. Wright, [0104]. In the combination, that policy includes Bittmann’s allowance for the account associated with the device, as explained in the motivation-to-combine section of Ground 1B. Bittmann, [0038], [0042]–[0043], [0045]. Allowance manager 230 updates that allowance from the rated reports, and when the allowance reaches its “pre-determined restriction level,” allowance manager 230 “causes the restriction of activity in the account.” *Id.*, [0043], [0045]. The allowance

and its restriction level thus define how much of a service the device may use under its policy.

415. The *specified condition* is the second alternative explained for 23[b]: either report shows restricted activity continuing after the allowance reached its restriction level and system 200 directed the restriction (*the specified condition comprises that either the first service usage indication or the second service usage indication is inconsistent with the service policy*). Restricted activity that continues after the allowance has reached its restriction level and system 200 has directed the restriction is usage that the device's policy does not permit, whichever report shows it. Bittmann, [0045], [0070]–[0071].

416. Upon detecting that condition, system 200 declares *a service processor integrity violation based on detecting the specified condition* as explained for 23[c]: it records the condition as a service processor integrity violation in the device's client management record. Wright, [0048], [0104].

417. Accordingly, Wright-Bittmann-Durham teaches that system 200 receives firmware agent 125's report (*the service usage indication is a first service usage indication*) and monitoring unit 130's report (*receiving a second service usage indication*) from client 201 (*from the service processor*), and that the condition system 200 detects (*the specified condition comprises*) is either report (*that either the first service usage indication or the second service usage indication*) showing

restricted activity continuing, after system 200 directed the restriction, beyond the restriction level of the allowance in the device's policy (*is inconsistent with the service policy*). Wright, [0104]; Bittmann, [0038], [0045], [0070]–[0071].

- d. Claim 30: “The method recited in claim 23, wherein the service usage indication comprises a start accounting message, a stop accounting message, a charging data record, a flow data record, a service usage measure, or a combination of these.”**

418. In my opinion, Wright-Bittmann-Durham teaches claim 30.

419. The *service usage indication* is firmware agent 125's trusted report, as explained for 23[a]. Claim 30 lists alternatives, and I understand that a service usage indication comprising any one of them satisfies the claim.

420. Because firmware agent 125 implements Bittmann's monitor, as explained in the motivation-to-combine section of this ground, its report includes the volume of the device's network traffic, a measure that Bittmann's reports include as the “volume of data transferred (e.g. kilobytes, number of packets, etc).” Bittmann, [0053], [0060]. That volume measures the device's use of network service (*comprises ... a service usage measure*): it quantifies, in kilobytes or packets, the data that the device transferred over the network, the same kind of quantity that Bittmann's allowances count as “volume in bytes” or “volume in packets.” *Id.*, [0038].

421. Accordingly, Wright-Bittmann-Durham teaches that firmware agent 125's report (*the service usage indication*) includes the volume of the device's network traffic (*comprises ... a service usage measure*), an alternative expressly recited in claim 30. Bittmann, [0053], [0060].

IX. SUMMARY AND ADDITIONAL REMARKS

422. It is my opinion that Grounds 1A, 1B, 1C and 1D each render the Challenged Claims unpatentable as obvious, as I explained in the paragraphs preceding.

423. I currently hold the opinions expressed in this declaration based on the information available to me at this time. I may, however, form additional or new opinions should additional information or insights be made available, or upon further consideration.

APPENDIX A

VITA

CHARLES D. CREUSERE

Klipsch School of Electrical & Computer Engineering

Mailing Address:

MSC 3-0

New Mexico State University

P.O. Box 30001

Las Cruces, NM 88003-8001

Phone: 575-646-3919

Fax: 575-646-1435

email: ccreuser@nmsu.edu

DISSERTATION TITLE

"Perfect Reconstruction Modulated Polyphase Filter Banks Using Reverse-Time Subfilters."

ACADEMIC TRAINING

1980-1985: University of California at Davis, B.S. in Electrical and Computer Engineering.

1989-1990: University of California at Santa Barbara, M.S. in Electrical and Computer Engineering.

1990-1993: University of California at Santa Barbara, Ph.D. in Electrical and Computer Engineering.

PROFESSIONAL EXPERIENCE

2010-current Holder of the Frank Carden Endowed Chair in Telemetry & Telecommunications (2010-2024) and Full Professor (current). Current research interests include compressive sensing/sparse reconstruction for LIDAR and streaming sensor data as well as EEG brain analysis for audiovisual perceptual quality assessment and modeling.

October 2008 Selected for the International Foundation for Telemetry Endowed Professorship.

Jan. 2000-2008 Assistant/Associate professor in the Klipsch School of Electrical & Computer Engineering. My teaching areas include digital signal processing, image processing, pattern classification, and source coding (signal compression). I have done past research in areas of image, video, and audio compression as well as feature vector extraction for pattern classification. Currently, my research interests include distributed compression, polarimetric image processing for scene analysis, and nonstationary signal denoising.

1993-1999: Researcher & Team Leader, Naval Air Warfare Center, China Lake. My research efforts have focused on high speed image and video compression technologies which offer unique capabilities such as robustness to transmission errors and regional localization. My team (2 other people) and I have implemented a real-time (3 to 15 frames/second with 240x512 frames)

320C80-based system which uses a wavelet transform along with embedded coding techniques to compress a video input and stream it through the Internet via TCP/IP protocols. Our recent research focus has been to add more intelligence to the encoder so that the space-frequency information in the image that is most useful for image analysis is received with the highest fidelity. While most of my recent research has been in the area of embedded compression, I am still very much interested in other applications of time/space-frequency decompositions and of multirate digital signal processing concepts in general.

1999, Spring Quarter: Instructor at the University of California at Santa Barbara. Taught graduate class in Multirate Digital Signal Processing, ECE 258B.

1990-1993: Research Assistant, Department of Electrical and Computer Engineering, University of California, Santa Barbara. Worked under Prof. S.K. Mitra on subband coding and multirate filter bank theory. Also implemented real-time filter banks on a Motorola 56001 digital signal processor.

1992: Summer Employee, AT&T Bell labs, Murray Hill, NJ. Developed and simulated new methods of extremely low bit rate video coding for video telephone applications.

1985-1989: Design Engineer, Naval Weapons Center, China Lake. Designed, built, and tested the guidance electronics for the Laser Guided Training Round. This project included mixed analog and digital circuit design as well as the programming of an embedded DSP. Also developed software for an advanced video processor and studied ground target tracking.

FUNDED RESEARCH

- (2000) Office of Naval Research, *Compression of Digital Elevation Maps Using Non-linear Wavelets*, 2000-2003, \$94K
- (2001) Sandia National Labs, *Intelligent Compression for Remote Sensing*, 2001-2003, \$70K.
- (2002) National Science Foundation (Early Career Grant), *Efficient Audio Compression with Perceptually Embedded Scalability*, 2002-2007, \$350K.
- (2004) National Geospatial-Intelligence Agency, *Passive Polarimetric Imagery Classification Study*, 2004-2006, \$160K (joint with Dr. David Voelz).
- (2005) Los Alamos National Laboratories, *Signal Detection via Adapted Filter Banks and Geometric Dimensionality Reduction*, 2005-2006, \$15K (unburdened).
- (2006) Los Alamos National Laboratories, *Signal Detection via Adapted Filter Banks and Geometric Dimensionality Reduction*, 2006-2007, \$50K (unburdened).
- (2006) National Geospatial-Intelligence Agency, *Exploiting Polarization in Imaging Systems*, 2006-2009, \$304K (joint with Dr. David Voelz).
- (2006) Army Research Office, *Distributed Source Coding Using Bitstream-based Detection and Classification*, 2006-2009, \$326K.
- (2006) DARPA (Subcontract from LANL), *ADAM Project*, 2006-2007, \$104K (joint with Dr. Joe Lakey and Dr. Jaime Ramirez)

- (2009) NMSU IRG, *Perceptual audio quality evaluation by direct measurement of human brain responses*, 2009-2010, \$39K (joint with Dr. Jim Kroger, Psychology)
- (2011) National Science Foundation, *CIF:Medium:Assessment and modeling of temporal variation in perceived audio and video quality using direct brainwave measurement*, 2011-2015, \$917K (lead PI with Dr. Jim Kroger and Dr. Joerg Kliewer as co-PIs)
- (2011) NASA EPSCOR, *Proximity Operations for Near Earth Asteroid Exploration*, 2011-2014, \$750K (co-PI, with Dr. Eric Butcher (lead), others)
- (2012) National Geospatial Intelligence Agency (NGA), *Pulse Complexity Based LIDAR Scene Modeling for Sparse Reconstruction and Super-Resolution*, 2012-2013 (plus 3 1 year options), \$150K (\$75K/option year), co-PI Dr. David Voelz.
- (2018) Airforce Research Lab (AFRL), *Software Radio Design in LabView FPGA*, 2018-2019, \$140K.

PATENTS

- Patent titled "Parallel digital image compression system which exploits zerotree redundancies in wavelet coefficients," Patent Number 6,148,111.
- Patent titled "Efficient embedded image and video compression using lifted wavelets," Number: 6,466,698, granted October 15, 2002.

OTHER DISTINCTIONS

- Awarded the International Foundation for Telemetering Professorship, October 2008.
- Received an educational fellowship from the Department of Defense, 1989-1992.
- Certificate of Merit for the outstanding technical paper awarded at the AIAA Missile Sciences Conference for the paper "Automatic target recognition directed image compression," Nov. 1998.
- Patent (classified) "Notice of Allowability" titled, "Microcontroller-Based Laser Pulse Decoder," granted October 7, 1991.
- Associate editor for IEEE Trans. on Image Processing, 2002-2005, 2010-2014
- Associate editor for IEEE Trans. on Multimedia , 2008-2013.
- Guest Editor, "Issue on Advances in Hyperspectral Data Processing and Analysis", IEEE Journal of Selected Topics in Signal Processing, Vol. 5, Numbers: 5 & 6, August-September 2015,
- Co-general chair, IEEE Digital Signal Processing Workshop, August 2004, Taos, NM.
- Co-technical chair for the 2012 and 2014 Southwest Symposium on Image Analysis and Interpretation.
- Student Paper Contest Chair, 40th Asilomar Conf. on Signals, Systems, and Computers, October 2006.

- Organized special session entitled "Applications of Multirate DSP" at the 40th Asilomar Conf. on Signals, Systems, and Computers, October 2006.
- Member of technical program committees for the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), the IEEE International Conference on Image Processing (ICIP), and the IEEE Data Compression Conference (DCC).
- Senior Area Editor, IEEE Transactions on Image Processing, March 2016 to May 2022.
- Area Editor for IEEE Access, June 2022-Present.
- Deputy Editor-in-Chief, IEEE Transactions on Image processing, May 2023-June 2024.

CONSULTING ACTIVITIES

- Video compression systems (technology consultant), Abba Tech, Albuquerque, NM, 2000.
- Expert witness in laser rangefinding technology, Asia Optical Inc. (through NY law firm of Osterlenk, Faber, Gerb & Soffen), Case: LTI versus Nikon/AOI, July 2001-2003. Case went to trial/ testified in court.
- Technical expert for defense; Case: Real-Time v. AT&T (byte.mobile), 2011-2012, case settled June 2012.
- Technical expert for defense; Case: Princeton Digital v. Dell, 2014-2015, case dismissed June 2015.
- Technical expert for defense; Noninfringement & IPR (6,597,812), Real-time v. SAP, 1/2016-6/2016.
- Technical expert for defense; IPRs (7,378,992 & 8,643,513), Real-time v. Riverbed, et al., 2/2016-2017
- Technical expert for defense; IPRs (7,415,530, 9,116,908, 7,161,506, & 9,054,728), Real-time v. Dell, et al., 2/2016-2017
- Technical expert for defense; Noninfringement, Real-time v. HP Enterprises, 4/2016-2018.
- Technical expert for defense; IPR (7,358,867, 7,161,506, & 9,054,728), Real-time v. Teradata 11/2016-2017
- Technical expert for defense; IPRs (7,415,530, 8,643,513 & 7,378,992), Real-time v. Veritas 12/2016-2017.
- Technical expert for defense; IPRs (7,075,917 & 6,304,612), UNILOC v. Apple 9/2018-2019.
- Technical expert for defense; IPR (7,558,730), Advanced Voice Recognition Systems v. Apple 7/2019-2020.
- Technical expert for defense; District Court; Noninfringement, Realtime Adaptive Systems v. YouTube/Google, 2018-2020.
- Technical expert for defense; ITC case; Nokia v. Lenovo, Oct. 2020-2021.
- Technical expert for defense; IPR (10,176,848), Maxell v. Apple, 2019-2021.

- Technical expert for plaintiff (rebutting invalidity); East Texas District Court; USAA v. PNC Bank, 2021-present. Deposed in Case 1 and Case 2; Attended trial in May 2022 for Case 1 and September 2022 for Case 2 but was not called to testify (USAA won on all counts, patents upheld as valid); I have also opined as an expert in 3 related IPRs (deposed in one of those cases, so far).
- Technical expert for defense; District Court; Gesture Tech Partners v. Apple/Lenovo&Motorola, 2021.

JOURNAL PUBLICATIONS

1. **C.D. Creusere and S.K. Mitra**, "A simple method for designing high-quality prototype filters for M-band pseudo-QMF banks," *IEEE Trans. on Signal Processing*, Vol. 43, No. 4, April 1995, pp. 1005-1007.
2. **C.D. Creusere and S.K. Mitra**, "Efficient audio coding using perfect reconstruction noncausal IIR filter banks," *IEEE Trans. on Speech and Audio Processing*, Vol. 4, No. 2, March 1996, pp. 115-123.
3. **C.D. Creusere and S.K. Mitra**, "Image coding using wavelets based on perfect reconstruction IIR filter banks," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, No. 5, Oct. 1996, pp. 447-458.
4. **C.D. Creusere**, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," *IEEE Trans. on Image Processing*, Vol 6, No. 10, Oct. 1997, pp. 1436-1442.
5. **C.D. Creusere and A. Van Nevel**, "ATR-directed image and video compression," *Journal of Aircraft*, Vol. 36, No. 4, pp. 626-31, July-August 1999.
6. **C.D. Creusere**, "Fast embedded compression for video," *IEEE Trans. on Image Processing*, Vol. 8, No. 12, pp. 1811-16, December 1999.
7. **C.D. Creusere**, "Motion compensated video compression with reduced complexity encoding for remote transmission," *Signal Processing: Image Communications*, Vol. 16, pp. 627-42, April 2000.
8. **C.D. Creusere**, "Understanding perceptual distortion in MPEG scalable audio coding," *IEEE Trans. on Speech and Audio Processing*, Vol. 13, No. 3, pp. 422-431, May 2005.
9. **L. E. Boucheron and C.D. Creusere**, "Lossless wavelet-based compression of digital elevation maps for fast and efficient search and retrieval," *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 43, No. 5, pp. 1210-1214, May 2005.
10. **V. Thilak, D. Voelz, and C.D. Creusere**, "Polarization-based index of refraction and reflection angle estimation for remote sensing applications," *Applied Optics*, Vol. 46, Bo. 30, pp. 7427-7536, Oct. 2007.
11. **C.D. Creusere, K. Kallakuri, and R. Vanam**, "An Objective Metric of Human Subjective Audio Quality Optimized for a Wide Range of Audio Fidelities," *Audio, Speech, and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]* , vol.16, no.1, pp.129-136, Jan. 2008
12. **S. Kandadai and C.D. Creusere**, "Scalable Audio Compression at Low Bitrates," *Audio, Speech, and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]* , vol.16, no.5, pp.969-979, July 2008
13. **S. Kandadai and C.D. Creusere**, "Reverse engineering and repartitioning vector quantizers using training set synthesis," *Signal Processing*, August 2008.
14. **V. Thilak, C.D. Creusere, and D. Voelz**, "Passive Polarimetric Imagery-Based Material Classification Robust to Illumination Source Position and Viewpoint," *Image Processing, IEEE Transactions on* , vol.20, no.1, pp.288-292, Jan. 2011.
15. **C.D. Creusere and J. Hardin**, "Assessing the Quality of Audio Containing Temporally Varying Distortions," *Audio, Speech, and Language Processing, IEEE Transactions on* , vol.19, no.4, pp.711-720, May 2011.

16. **Castorena, J.; Creusere, C.D.**, "The Restricted Isometry Property for Banded Random Matrices," *Signal Processing, IEEE Transactions on* , vol.62, no.19, pp.5073-5084, Oct.1, 2014
doi: 10.1109/TSP.2014.2345350.
17. **Castorena, J.; Creusere, C.D.**, "Sampling of Time-Resolved Full-Waveform LIDAR Signals at Sub-Nyquist Rates," *Geoscience and Remote Sensing, IEEE Transactions on* , vol.53, no.7, pp.3791-3802, July 2015. doi: 10.1109/TGRS.2014.2383839.

REFEREED CONFERENCE PUBLICATIONS

1. **H. Babic, S.K. Mitra, C.D. Creusere, and A. Das**, "Perfect reconstruction recursive QMF banks for image subband coding," *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1991, pp. 746-750.
2. **S.K. Mitra, C.D. Creusere, and H. Babic**, "A novel implementation of perfect reconstruction QMF banks using IIR filters," *Proc. IEEE Int. Symposium on Circuits and Systems*, San Diego, CA, May 1992, pp. 2312-2315.
3. **S.K. Mitra, C.D. Creusere, and H. Babic**, "Design of transmultiplexers using IIR filter banks," *Signal Processing VI: Theories and Applications*, Elsevier Science Publishers, 1992, pp. 223-226.
4. **C.D. Creusere and S.K. Mitra**, "Efficient image scrambling using polyphase filter banks," *Proc. International Conference on Image Processing*, Austin, TX, Nov. 1994, pp. 81-85.
5. **C.D. Creusere and G. Hewer**, "Wavelet-based nearest neighbor pattern classification using scale sequential matching," *Proc. Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1994, pp. 1123-1127.
6. **C.D. Creusere**, "Embedded zerotree image coding using low complexity IIR filter banks," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, May 1995, pp. 2213-16.
7. **C.D. Creusere and Gary Hewer**, "Digital video compression for weapons control and bomb damage indication," *AGARD Conference Proceedings 576*, Chapter 16, Sept. 1995.
8. **C.D. Creusere**, "Image coding using parallel implementations of the embedded zerotree wavelet algorithm," *Proc. of the Digital Video Compression Conference (Algorithms and Technologies 1996)*, San Jose, CA, Jan. 28-Feb. 2, 1996, pp. 82-93.
9. **C.D. Creusere**, "A family of image compression algorithms which are robust to transmission errors," *Proceedings of the SPIE*, Vol. 2825, Denver, CO, August, 1996, pp. 890-900.
10. **C.D. Creusere**, "Perfect reconstruction time-varying IIR filter banks," *Conf. Rec. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1996, pp. 1319-23.
11. **C.D. Creusere**, "Out-of-loop motion compensation for reduced complexity video encoding," *Proc. of the Data Compression Conf.* (pp. 428) & *Data Compression Industry Workshop* (pp.28-37), March 1997, Snowbird, UT.
12. **C.D. Creusere**, "Periodic pan compensation for reduced complexity video compression," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol IV, pp. 2889-92, April 1997, Munich, Germany.
13. **C.D. Creusere**, "A new approach to global motion compensation which reduces video encoding complexity," *Proc. Int. Conf. on Image Processing*, Vol. III, pp. 634-7, October 1997, Santa Barbara, CA.
14. **C.D. Creusere**, "Spatially partitioned lossless image compression in an embedded framework," *Conf. Rec. 31st Asilomar Conf. on Signals, Systems, and Computers*, Nov. 1997, Pacific Grove, CA.
15. **C.D. Creusere**, "Adaptive embedding for reduced complexity image and video compression," *Proc.*

- of the *SPIE*, Vol 3309 (Visual Communications and Image Processing), pp. 48-57, Jan. 1998, San Jose, CA.
16. **C.D. Creusere**, "Successive coefficient refinement for embedded lossless image compression," *Proc. of the Data Compression Conf.*, pp. 539, March 1998, Snowbird, UT.
 17. **C.D. Creusere**, "Subband coding of speech and audio," *Proc. of the European Signal Processing Conf.* (invited paper), Sept. 1998, Isle of Rhodes, Greece.
 18. **C.D. Creusere**, "Fast embedded video compression using cache-based processing," *Proc. of the European Signal Processing Conf.*, Sept. 1998, Isle of Rhodes, Greece.
 19. **C.D. Creusere**, "Successive coefficient refinement for embedded lossless image compression," *Proc. Int. Conf. on Image Processing*, Oct. 1998, Chicago, IL.
 20. **C.D. Creusere** and A. Van Nevel, "Autonomous target recognition directed image compression," *Proc. of the AIAA*, Nov. 1998.
 21. **C.D. Creusere**, "Improved successive refinement for wavelet-based embedded image compression," *Proc. of the SPIE*, Denver, CO, July 1999.
 22. A. Van Nevel and **C.D. Creusere**, "Intelligent Bandwidth Compression," *Proc. of the SPIE*, Denver, CO, July 1999.
 23. **C.D. Creusere**, "Optimal refinement/significance map tradeoffs in SPIHT-based image compression," *Conf. Rec., 34th Asilomar Conf. on Signals, Systems, & Computers*, pp. 1026-30, Oct. 2000.
 24. **C.D. Creusere**, "Compression of digital elevation maps using nonlinear wavelets," *Proc. Int. Conf. on Image Processing*, pp. 824-7, October 2001.
 25. **C.D. Creusere and G. Dahman**, "Object detection and localization in compressed video," *Conf. Rec. 35th Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2001, Pacific Grove, CA.
 26. **C.D. Creusere**, "An analysis of perceptual artifacts in MPEG scalable audio coding," *Proceedings of the Data Compression Conference*, pp. 152-161, April 2002, Snowbird, UT.
 27. **C.D. Creusere and N. Tolk**, "Combining wavelets and GLICBAWLS to achieve resolution-progressive lossless compression," *Proc. of the International Conference on Image Processing*, pp. III-229-32, October 2002.
 28. **L. Boucheron and C.D. Creusere**, "Compression of digital elevation maps for fast and efficient search and retrieval," *Proc. of the International Conference on Image Processing*, pp. 629-32, September 2003.
 29. **S. Kandadai and C.D. Creusere**, "An experimental study of object detection in the wavelet domain," *Conf. Rec. 37th Asilomar Conf. on Signals, Systems, and Computers*, pp. 1620-4, Nov. 2003, Pacific Grove, CA.
 30. **C.D. Creusere**, "Quantifying perceptual distortion in scalably compressed MPEG audio," *Conf. Rec. 37th Asilomar Conf. on Signals, Systems, and Computers*, pp. 265-9, Nov. 2003, Pacific Grove, CA.
 31. **C.D. Creusere and L. Zhou**, "Spatial object detection and classification in JPEG bitstreams," *Proceedings 11th Digital Signal Processing Workshop*, pp. 115-9, August 2004, Taos Ski Valley, NM.
 32. **L. Zhou and C.D. Creusere**, "Spatial object detection in JPEG bitstreams," *Proceedings European Conference on Signal Processing*, pp. 949-52, September 2004, Vienna, Austria.
 33. **V. Thilak and C.D. Creusere**, "Tracking of extended size targets in H.264 compressed video using the probabilistic data association filter," *Proceedings European Conference on Signal Processing*, pp. 281-4, September 2004, Vienna, Austria.
 34. **S. Kandadai and C.D. Creusere**, "Reverse engineering vector quantizers by training set synthesis," *Proceedings European Conference on Signal Processing*, pp. 789-92, September 2004, Vienna, Austria.

35. **V. Thilak, J. Saini, D. G. Voelz, C. D. Creusere** , "Pattern recognition for passive polarimetric data using nonparametric classifiers," *Proc. of the SPIE Vol. 5888*, p. 337-344, Polarization Science and Remote Sensing II; Joseph A. Shaw, J. Scott Tyo; Eds.
36. **R. Vanam and C.D. Creusere**, "Evaluating low bitrate scalable audio quality using advanced version of PEAQ and energy equalization approach ," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. III, pp. 189-92, March 2005.
37. **S. Kandadai and C.D. Creusere**, "Reverse engineering vector quantizers for repartitioned signal spaces," *Proc. 39th Asilomar Conference on Signals, Systems, and Computers*, pp. 1208-12, Pacific Grove, CA, Nov. 2005.
38. **R. Vanam and C.D. Creusere**, "Scalable perceptual metric for evaluating audio quality," *Proc. 39th Asilomar Conference on Signals, Systems, and Computers*, pp. 319-23, Pacific Grove, CA, Nov. 2005.
39. **S. Kandadai and C.D. Creusere**, "Perceptually-weighted audio coding that scales to extremely low bitrates," *Proc. IEEE Data Compression Conference*, pp. 382-391, Snowbird, UT, March 2006.
40. **V. Thilak, D. Voelz, C. Creusere, S. Damarla**, "Estimating the refractive index and reflected zenith angle of a target using multiple polarization measurements," *Proc. SPIE Vol. 6240, 624004*, Polarization: Measurement, Analysis, and Remote Sensing VII; Dennis H. Goldstein, David B. Chenault; Eds., May 2006.
41. **V.M. Prasad and C.D. Creusere**, "Analyzing reversible lapped transformations using Reng probing," *Proc. 40th Asilomar Conference on Signals, Systems, and Computers*, pp. 873-877, Oct. 2006.
42. **N. Balachandran and C.D. Creusere**, "Chirp classification using hidden Markov models," *Proc. 40th Asilomar Conference on Signals, Systems, and Computers*, pp. 545-549, Oct. 2006.
43. **V. Thilak and C.D. Creusere**, "Estimating the complex index of refraction and view angle of an object using multiple polarization measurements," *Proc. 40th Asilomar Conference on Signals, Systems, and Computers*, pp. 1067-1071, Oct. 2006.
44. **V. Thilak, C.D. Creusere, and D.G. Voelz**, "Material classification using passive polarimetric imagery," *Proc. IEEE Int. Conf. on Image Proc.*, Vol. 4, pp. IV-121-124, Sept., 2007.
45. **A. Pamba, V. Thilak, D. G. Voelz, C. D. Creusere** "Estimation of incidence and reflection angles from passive polarimetric imagery: extension to out-of-plane scattering," *Proc. SPIE Vol 6682, 668200*, Polarization Science and Remote Sensing III, Joseph A. Shaw; J. Scott Tyo, Editors, September 2007.
46. **V. Thilak, D. G. Voelz, C. D. Creusere**, "Image segmentation from multi-look passive polarimetric imagery," *Proc. SPIE 6682, 668206*, Polarization Science and Remote Sensing III, Joseph A. Shaw; J. Scott Tyo, Editors, October 2007.
47. **S. Kandadai and C.D. Creusere**, "Optimal Bit Layering for Scalable Audio Compression Using Objective Audio Quality Metrics," *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on* , pp.560-564, 4-7 Nov. 2007.
48. **C.D. Creusere and I. Mecimore**, "Bitstream-based overlap analysis for multi-view distributed video coding," *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 93-96, March 2008.
49. **V. Thilak, C.D. Creusere, and D.G. Voelz**, "Passive Polarimetric Imagery Based Material Classification For Remote Sensing Applications," *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 153-156 March 2008.
50. **V. Thilak, Q. Wang, D. G. Voelz, C. D. Creusere**, "Estimation of target geometry from Mueller matrix imagery," *Proc. SPIE 6972*, March 2008.
51. **C.D. Creusere and I. Mecimore**, "Bitstream-based correlation detector for multi-view distributed video coding applications," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp.

1001-1004, April 2008, Las Vegas, NV.

52. **S. Kandadai, J. Hardin, and C.D. Creusere**, "Audio quality assessment using the mean structural similarity measure," *Proc. Int. Conf. on Accoustics, Speech, and Signal Processing*, pp. 221-224, April 2008, Las Vegas, NV.
53. **S. Matta and C.D. Creusere**, "Efficient correlation extraction for distributed audio coding," *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, vol., no., pp.1272-1276, 26-29 Oct. 2008.
54. **J.C. Hardin and C.D. Creusere**, "Objective analysis of temporally varying audio quality metrics," *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, vol., no., pp.1245-1249, 26-29 Oct. 2008.
55. **S. Matta and C.D. Creusere**, "Distributed audio coding with efficient source correlation extraction," *Proceedings 13th Digital Signal Processing Workshop*, pp. 16-20, January 2009, Marco Island, FL.
56. **I. Mecimore and C.D. Creusere**, "Unsupervised bitstream based segmentation of images," *Proceedings 13th Digital Signal Processing Workshop*, pp. 642-647, January 2009, Marco Island, FL.
57. **Q. Wang, C.D. Creusere, V. Thilak, and D.G. Voeltz**, "Active polarimetric imaging for estimation of scene geometry," *Proceedings 13th Digital Signal Processing Workshop*, pp. 659-663, January 2009, Marco Island, FL.
58. **J.C. Hardin and C.D. Creusere**, "A temporally varying objective audio quality metric," *Proceedings 13th Digital Signal Processing Workshop*, pp. 21-25, January 2009, Marco Island, FL.
59. **I. Mecimore, W. Fahrenkrog, and C.D. Creusere**, "On bitstream based edge detection techniques," ITEA Conference, January 2009, El Paso, TX (best graduate student paper award).
60. **Mecimore, Ivan; Creusere, Charles D.**, "Low complexity multi-view distributed video coding based on JPEG," *Image Analysis & Interpretation (SSIAI), 2010 IEEE Southwest Symposium on*, vol., no., pp.165-168, 23-25 May 2010.
61. **Creusere, C.D.; Mehta, K.; Voeltz, D.G.**, "Model-based estimation of surface geometry using passive polarimetric imaging," *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, vol., no., pp.4557-4560, 25-30 July 2010.
62. **Castorena, J.; Creusere, C.D.; Voeltz, D.**, "Modeling lidar scene sparsity using compressive sensing," *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, vol., no., pp.2186-2189, 25-30 July 2010.
63. **Castorena, J.; Creusere, C.D.; Voeltz, D.**, "Using finite moment rate of innovation for LIDAR waveform complexity estimation," *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, vol., no., pp.608-612, 7-10 Nov. 2010.
64. **Creusere, C.D.; Siddenki, S.; Hardin, J.; Kroger, J.**, "Early investigations into subjective audio quality assessment using brainwave responses," *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, vol., no., pp., Nov. 2011.
65. **Castorena, J.E.; Creusere, C.**, "Remote-sensed LIDAR using random sampling and sparse reconstruction," *Proc. International Telemetering Conference*, Las Vegas, NV, October 2011.
66. **Davis, P; Creusere, C.**, "Quantifying the gains of compressive sensing for telemetering application," *Proc. International Telemetering Conference*, Las Vegas, NV, October 2011.
67. **Creusere, C.D.; Kroger, J.; Siddenki, S.R.; Davis, P.; Hardin, J.**, "Assessment of subjective audio quality from EEG brain responses using time-space-frequency analysis," *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, vol., no., pp.2704-2708, 27-31 Aug. 2012.
68. **Castorena, J.; Creusere, C.D.**, "Compressive sampling of LIDAR: Full-waveforms as signals

of finite rate of innovation," *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European* , vol., no., pp.984-988, 27-31 Aug. 2012.

69. **Castorena, J.; Creusere, C.D.**; "Random impulsive scan for LIDAR sampling," *Proc. IEEE Int. Conf. on Image Proc.*, October 2012.

70. **Creusere, C., Nelson, E., Critz, T., Butcher, E.**; "Analysis of communication interconnectedness in the proximity of near-earth asteroids," *Proc. International Telemetering Conference*, San Diego, CA, October 2012.

71. **Castorena, J.E.; Creusere, C.**; "Remote-sensed LIDAR using random impulsive scans," *Proc. International Telemetering Conference*, San Diego, CA, October 2012.

72. **Castorena, J.; Creusere, C.D.**, "Sub-spot localization for spatial super-resolved LIDAR," *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* , vol., no., pp.2227,2231, 26-31 May 2013.

73. **Charles D. Creusere ; Juan Castorena**, "A unified framework for 3rd generation lidar pulse processing based on finite rate of innovations," *Proc. SPIE 8858, Wavelets and Sparsity XV*, 88580T (September 26, 2013); doi:10.1117/12.2022447.

74. **Nelson, E., Creusere, C., Critz, T., Butcher, E.**; "Analysis of communication rates in the proximity of near-earth asteroids," *Proc. International Telemetering Conference*, Las Vegas, NV, October 2013.

75. **Castorena, J.E.; Creusere, C.**; "Full-waveform LIDAR recovery at sub-Nyquist rates," *Proc. International Telemetering Conference*, Las Vegas, NV, October 2013.

76. **Davis, P.; Creusere, C.D.; Kroger, J.**, "EEG and the human perception of video quality: Impact of channel selection on discrimination," *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE* , vol., no., pp.9,12, 3-5 Dec. 2013, doi: 10.1109/GlobalSIP.2013.6736798.

77. **Creusere, C.D.; McRae, N.; Davis, P.**, "Sample-based cross-frequency coupling analysis with CFAR detection," *Signals, Systems and Computers, 2014 48th Asilomar Conference on* , vol., no., pp.179,183, 2-5 Nov. 2014; doi: 10.1109/ACSSC.2014.7094423.

78. **Davis, P.; Creusere, C.D.; Kroger, J.**, "Classification of human viewers using high-resolution EEG with SVM," *Signals, Systems and Computers, 2014 48th Asilomar Conference on* , vol., no., pp.184,188, 2-5 Nov. 2014; doi: 10.1109/ACSSC.2014.7094424.

79. **Davis, P.; Creusere, C. D.; Tang, W.**, "ASIC implementation of the cross frequency coupling algorithm for EEG signal processing," *2014 14th International Symposium on Integrated Circuits (ISIC)*, pp.248-251, Dec. 2014 doi: 10.1109/ISICIR.2014.7029468

80. **Nelson, Evan; Creusere, Charles D.; Butcher, Eric**, "Determining position around an asteroid using communication relays and trilateration," *Aerospace Conference, 2015 IEEE* , vol., no., pp.1,6, 7-14 March 2015; doi: 10.1109/AERO.2015.7118955.

81. **Davis, Philip; Creusere, Charles D.; Tang, Wei**, "Window length effect on cross frequency coupling in an EEG processing circuit," in *Circuits and Systems (MWSCAS), 2015 IEEE 58th International Midwest Symposium on* , vol., no., pp.1-4, 2-5 Aug. 2015 doi: 10.1109/MWSCAS.2015.7282125.

82. **Davis, Philip; Creusere, Charles D.; Kroger, Jim**, "Subject Identification Based on EEG Response to Video Stimuli," *Proceedings International Conference on Image Processing, 2015 IEEE*, September 2015.

83. **Newtonson, Kathy; Creusere, Charles D.**, "Histogram oriented gradients and map seeking circuits pattern recognition with compressed imagery," *Proc. Southwest Symposium on Image Analysis and Interpretation*, Santa Fe, NM, March 2016.

84. **Davis, Philip; Creusere, Charles D.; Kroger, Jim**, "Assessing Cross Frequency Coupling in EEG Collected from Subjects Viewing Video using a Modified Metric," *Proc. Southwest Symposium on Image Analysis and Interpretation*, Santa Fe, NM, March 2016.

85. **Davis, Philip; Creusere, Charles D.; Kroger, Jim**, "The Effect of Perceptual Video Quality on EEG Power Distribution," *Proceedings International Conference on Image Processing, 2016 IEEE*, Phoenix, AZ, September 2016.
86. **Creusere, Charles D.; Castorena, J.; Dragulin, I.; Voelz, D.**, "Quantifying the Accuracy of FRI based LIDAR Waveform Analysis," *Proceedings International Conference on Image Processing, 2016 IEEE*, Phoenix, AZ, September 2016.
87. **Kathy A. Newton, Charles D. Creusere**, "Compressed imagery detection rate through map seeking circuit, and histogram of oriented gradient pattern recognition", *Proc. SPIE 10203, Pattern Recognition and Tracking XXVIII*, 102030F (1 May 2017); doi:10.1117/12.2262919; <https://doi.org/10.1117/12.2262919>
88. **Erandi Wijerathna, Charles D. Creusere, David Voelz, Juan Castorena**, "Polarimetric LIDAR with FRI sampling for target characterization", *Proc. SPIE 10407, Polarization Science and Remote Sensing VIII*, 104070R (7 September 2017); doi: 10.1117/12.2272587; <https://doi.org/10.1117/12.2272587>
89. **Andrew J. Phillips, Charles D. Creusere**, "The effects of lossy EEG compression on ERP analysis," *International Telemetering Conference Proceedings*, November 2018, <http://hdl.handle.net/10150/631654>.
90. **Andrew J. Phillips, Charles D. Creusere**, "The effects of lossy frequency-domain EEG compression on cross-frequency coupling analysis," *International Telemetering Conference Proceedings*, October 2019.
91. **Sergio Lara, Charles D. Creusere**, "Software Defined Radio for Carrier and Symbol Timing Extraction," *International Telemetering Conference Proceedings*, October 2021.
92. **Sergio Lara, Charles D. Creusere**, "LabView for Software Defined Radio Development," *International Telemetering Conference Proceedings*, October 2021.
93. **Ayatollah Elsayed, Charles D. Creusere**, "The Feedback Dynamics of Brain-Computer Interfaces in a Distributed Processing Environment," *Proceedings of HICSS-55*, January 2022; <http://hdl.handle.net/10125/79821>.
94. **Joshua Tellez, Charles D. Creusere**, "Evaluation of JPEG-2000 Image Compression as Applied to Electroencephalography," *International Telemetering Conference Proceedings*, Glendale, AZ, October 2022, <https://repository.arizona.edu/handle/10150/666964>.

PRESENTATIONS & OTHER PUBLICATIONS

1. **C.D. Creusere**, "Generic Tracking Laboratory Study," *NWC Technical Paper 7018*, Nov. 1989, Naval Weapons Center, China Lake, CA.
2. **C.D. Creusere**, "Robust image coding using the embedded zerotree wavelet algorithm," presented at the Digital Compression Conference, Snowbird, UT, April 1996.
3. **C.D. Creusere**, "Creating a compressed and embedded image representation which is robust to transmission errors," presented at the Office of Naval Research Workshop on Error Resilient Compression, San Diego, CA, February, 1996.
4. **C.D. Creusere**, "Data Compression Project Final Report," NAWCWD Technical Paper, TP 8442.
5. **C.D. Creusere**: "Applications of multirate digital signal processing to communications systems," NATO/RTA lecture series 216 on *Applications of Mathematical Signal Processing Techniques to Mission Systems*, presented in Paris, France; Cologne, Germany; and Monterey, CA,

November 1999.

6. **C.D. Creusere**, "Object detection and recognition in compressed video," invited presentation and white paper for the Motion Imagery Workshop, sponsored by the National Imagery and Mapping Agency (NIMA) and the National Science Foundation (NSF).
7. **C.D. Creusere**, "Application specific compression: digital elevation map browsing, video surveillance, and scalable audio," invited seminar at the University of Arizona, Tucson, AZ, March 2003.
8. **C.D. Creusere**, "Compressing Digital Elevation Maps for Efficient Search and Retrieval," presented at the Office of Naval Research Workshop on Image Processing, Minneapolis, MN, May 2003.