

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Gregory G. Raleigh Attorney Docket No.39843-0190IP1
U.S. Patent No.: 10,028,144
Issue Date: July 17, 2018
Appl. Serial No.: 14/948,065
Filing Date: November 20, 2015
Title: SECURITY TECHNIQUES FOR DEVICE ASSISTED SERVICES

Mail Stop Patent Board

Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES
PATENT NO. 10,028,144 PURSUANT TO 35 U.S.C. §§ 311–319,
37 C.F.R. § 42

TABLE OF CONTENTS

I.	IPR REQUIREMENTS	1
A.	Standing	1
B.	Challenge, Relief Requested.....	1
C.	Claim Construction	2
II.	'144 Patent	2
III.	THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	3
A.	GROUND 1A: Wright Renders Obvious Claims 1-11, 13, 15, and 18 ..	3
1.	Wright.....	3
2.	Claim Analysis	6
B.	[GROUND 1B] – Wright in View of Tzannes Renders Obvious Claims 12 and 16-17	72
1.	Tzannes.....	72
2.	Combination of Wright and Tzannes	74
3.	Claim Analysis	78
C.	[GROUND 1C] – Wright in View of Smith Renders Obvious Claim 14	83
1.	Smith.....	83
2.	Combination of Wright and Smith.....	84
3.	Claim 14	87
IV.	TRIAL SHOULD BE INSTITUTED.....	87
V.	CONCLUSION AND FEES	88
VI.	MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(a)(1).....	88
A.	Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1).....	88
B.	Related Matters Under 37 C.F.R. § 42.8(b)(2)	88
C.	Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3).....	88
D.	Service Information	89

EXHIBITS

SAMSUNG-1001	U.S. Patent No. 10,028,144 to Gregory G. Raleigh (“the ’144 Patent”)
SAMSUNG-1002	Excerpts from the Prosecution History of the ’144 Patent
SAMSUNG-1003	Declaration and Curriculum Vitae of Dr. Patrick McDaniel
SAMSUNG-1004	Complaint, <i>Headwater Research LLC v. Samsung Electronics Co., Ltd. et al</i> , 2-24-cv-00627, E.D. Tex., filed August 2, 2024
SAMSUNG-1005	U.S. Patent Application No 2008/0052395 A1 (“Wright”)
SAMSUNG-1006	U.S. Patent Application No. 2009/0300450 A1 (“Tzannes”)
SAMSUNG-1007	(Excerpts) Smith, et al., 2005. “Virtual Machines: Versatile Platforms for Systems and Processes,” Elsevier, Inc, 2005, ISBN 1-55860-910-5 (“Smith”)
SAMSUNG-1008	RESERVED
SAMSUNG-1009	U.S. Patent Application No. 2005/0144505 A1 (“Takeuchi”)
SAMSUNG-1010	U.S. Patent Application No. 2009/0254465 A1 (“Oester”)
SAMSUNG-1011	U.S. Patent Application No. 2006/0015749 A1 (“Mittal”)
SAMSUNG-1012	Declaration of June Munford
SAMSUNG-1013	Mallick, M., <i>Mobile and Wireless Design Essentials</i> (2003)
SAMSUNG-1014	Anderson, R., <i>Security Engineering: A Guide to Building Dependable Distributed Systems</i> . (2001)
SAMSUNG-1015	Jaeger, T., <i>Operating System Security (Synthesis Lectures on Information Security, Privacy, and Trust)</i> (2008)

SAMSUNG-1016	Parziale, L., TCP/IP Tutorial and Technical Overview (IBM Redbooks) (2006)
SAMSUNG-1017	McCanne, S. et al., The BSD Packet Filter: A New Architecture for User-level Packet Capture (1992)
SAMSUNG-1018	Horak, R., <i>Webster's New World Telecom Dictionary</i> , (2008)
SAMSUNG-1019	Ganz, A., et al., Multimedia Wireless Networks: Technologies, Standards, and QoS (2003)
SAMSUNG-1020	Kwok, Y., et al., Wireless Internet and Mobile Computing: Interoperability and Performance (2007)
SAMSUNG-1021	Brownlee et al., Traffic Flow Measurement: Architecture (1997)
SAMSUNG-1022	Saltzer, J. H., The Protection of Information in Computer Systems, Proceedings of the IEEE (1975)
SAMSUNG-1023	Tan, et al., An Empirical Study on the Capacity and Performance of 3G Networks (2008)
SAMSUNG-1024	Wang, et al., Shield: Vulnerability-Driven Network Filters for Preventing Known Vulnerability Exploits, ACM SIGCOMM Computer Communication Review (2004)
SAMSUNG-1025	Silberschatz et al., Operating System Concepts (7th ed. 2005)
SAMSUNG-1026	Cheswick et al., Firewalls and Internet Security (2nd ed. 2003)
SAMSUNG-1027	Koren et al., Fault Tolerant Systems (2007)
SAMSUNG-1028	Loscri et al., Voice and Video Telephony Services in Smartphone, EURASIP Journal on Wireless Communications and Networking (2006)
SAMSUNG-1029	Oxford University Press, <i>A Dictionary of Computing</i> , (6th ed. 2008) (defining "Agent")

SAMSUNG-1030	Downing et al., <i>Dictionary of Computer and Internet Terms</i> (9th ed. 2006) (defining “Agent”)
SAMSUNG-1031	Kunie et al., "Low power architecture and design techniques for mobile handset LSI Medity M2," 748-753, 2008 Asia and South Pacific Design Automation Conference (2008).
SAMSUNG-1032	M. Naruse et al., "A 65nm Single-Chip Application and Dual-Mode Baseband Processor with Partial Clock Activation and IP-MMU," 260-612, 2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers (2008).
SAMSUNG-1033	Gertz et al., <i>Handbook of Database Security: Applications and Trends</i> , Springer (2007).
SAMSUNG-1034	Oney, <i>Programming The Microsoft Windows Driver Model</i> (2nd ed. 2003).
SAMSUNG-1035	Exhibit A, <i>Headwater Research LLC v. Samsung Electronics Co., Ltd. et al</i> , 2-24-cv-00627, E.D. Tex., filed August 2, 2024.
SAMSUNG-1036	Patent Trial and Appeal Board (PTAB) Boardside Chat: Interim processes relating to institution in AIA proceedings, downloaded from https://www.uspto.gov/sites/default/files/documents/boardside_chat_interim_process_for_aia_institution_decisions_.pdf on April 25, 2025
SAMSUNG-1037	Samsung Stipulation Letter
SAMSUNG-1038	Interim Process for PTAB Workload Management
SAMSUNG-1039	RESERVED
SAMSUNG-1040	IEEE 100, <i>The Authoritative Dictionary of IEEE Standards Terms</i> (7th ed. 2000)
SAMSUNG-1041	U.S. Patent No. 7,962,798 to Locasto et al. (“Locasto”)

SAMSUNG-1042	Tanenbaum et al., Modern Operating Systems (2001)
SAMSUNG-1043	U.S. Patent Application No. 2007/0143827 A1 (“Nicodemus”)
SAMSUNG-1044	U.S. Patent Application No. 2007/0006282 A1 (“Durham”)
SAMSUNG-1045	U.S. Patent No. 7,633,909 B1 (“Jones”)
SAMSUNG-1046	U.S. Patent No. 9,032,192B2 (“Frank”)
SAMSUNG-1047	The Secure Sockets Layer (“SSL”) Protocol, V. 3.0, available at https://web.archive.org/web/19970614041044/http://home.netscape.com/eng/ssl3/ssl-toc.html and https://web.archive.org/web/19970617034012/http://home.netscape.com/eng/ssl3/3-SPEC.HTM#1

CLAIMS

Claim	Identifier	Claim Language
1	[1.1]	A wireless end-user device comprising:
	[1.2]	a wireless wide area network (WWAN) modem to receive and transmit Internet data between the device and at least one WWAN when configured for and connected to the at least one WWAN; and
	[1.3]	one or more processors configured to execute one or more processes in a kernel execution partition, one or more processes in an application execution partition, and one or more processes in a protected execution partition,
	[1.4]	the one or more processes executed in the kernel execution partition including an operating system packet network stack to pass Internet data packet traffic between the WWAN modem and one or more applications executing in the application execution partition,
	[1.5]	the one or more processes executed in the kernel execution partition including ... an application identification agent to classify individual flows of the Internet data packet traffic passing through the stack according to which of the one or more applications is associated with a particular individual flow,
	[1.6]	the one or more processes executed in the kernel execution partition including ... a service measurement agent to measure, for classified individual flows, an amount of the Internet data packet traffic associated with each of the one or more applications, and
	[1.7]	] the one or more processes executed in the kernel execution partition including ... a policy

Claim	Identifier	Claim Language
		control agent to apply application-specific traffic policy controls to classified individual flows,
	[1.8]	the one or more processes executed in the protected execution partition including one or more device agents with limited privileges to access processes executing in the kernel execution partition, including
	[1.9]	one or more device agents ... including a privilege to configure the application-specific traffic policy controls of the policy control agent, and
	[1.10]	one or more device agents ... including ... a privilege to receive at least one of Internet data packet traffic passing between the operating system packet network stack and the WWAN modem, and traffic information including the measured amount of the Internet data packet traffic associated with each of the one or more applications.
2	[2]	The wireless end-user device of claim 1, the one or more processors further configured to execute, in the application execution partition, at least one device-assisted-services agent to communicate with at least one device agent executed in the protected execution partition.
3	[3.1]	The wireless end-user device of claim 1, further comprising one or more memory partitions to store program code for the respective processes of the operating system packet network stack, the application identification agent, the service measurement agent, the policy control agent, and the one or more device agents, the one or more processors configured to retrieve the stored program code from the one or more

Claim	Identifier	Claim Language
		memory partitions to execute the respective processes.
	[3.2]	The wireless end-user device of claim 1, further comprising one or more memory partitions to store program code for the respective processes of the operating system packet network stack, the application identification agent, the service measurement agent, the policy control agent, and the one or more device agents, the one or more processors configured to retrieve the stored program code from the one or more memory partitions to execute the respective processes.
4	[4.1]	The wireless end-user device of claim 3, wherein the one or more memory partitions include a separately protected device-assisted services partition to store the program code for the one or more device agents,
	[4.2]	the one or more device agents further comprising an access control integrity agent having exclusive permission to update program code stored in the separately protected device-assisted services partition.
5	[5]	The wireless end-user device of claim 4, the access control integrity agent further configured to validate the program code for the one or more device agents.
6	[6]	The wireless end-user device of claim 3, wherein the one or more memory partitions comprise a device-assisted services partition to store device service state information, the device service state information only modifiable by the one or more device agents.

Claim	Identifier	Claim Language
7	[7]	The wireless end-user device of claim 6, the one or more device agents further comprising a service control link configured to securely communicate with a network service controller.
8	[8]	The wireless end-user device of claim 7, wherein the service control link is further configured to receive at least one of the application-specific traffic policy controls from the network service controller.
9	[9]	The wireless end-user device of claim 7, wherein the service control link is further configured to transmit at least some of the device service state information to the network service controller.
10	[10]	The wireless end-user device of claim 9, wherein the one or more device agents are further configured to retain the transmitted device service state information until after an instruction is received from the network service controller acknowledging that the device service state information can be deleted.
11	[11]	The wireless end-user device of claim 1, the one or more device agents further configured to, based on the application-specific traffic policy controls, configure application settings for the one or more applications.
12	[12]	The wireless end-user device of claim 1, the one or more device agents further configured to calculate, separate from the traffic information, a service usage measurement for the Internet data packet traffic passing between the operating system packet network stack and the WWAN modem.
13	[13]	The wireless end-user device of claim 1, the one or more device agents further configured to encrypt Internet data packet traffic passing

Claim	Identifier	Claim Language
		from the operating system packet network stack to the WWAN modem.
14	[14]	The wireless end-user device of claim 1, wherein the protected execution partition comprises a secure virtual execution environment.
15	[15]	The wireless end-user device of claim 1, wherein the privilege to configure the application-specific traffic policy controls of the policy control agent is an exclusive privilege for one or more of the device agents executed in the protected execution partition.
16	[16]	The wireless end-user device of claim 1, the WWAN modem comprising a modem execution partition configured to calculate, separate from the traffic information, a service usage measurement for the Internet data packet traffic passing between the operating system packet network stack and the WWAN modem.
17	[17]	The wireless end-user device of claim 16, the modem execution partition further comprising a modem control link to communicate the calculated service usage measure to at least one of a network service controller and at least one device agent executed in the protected execution partition.
18	[18]	The wireless end-user device of claim 2, further comprising a user interface, the at least one device-assisted-services agent comprising a user-interface agent to display to a user, and allow modification to, at least one policy setting affecting at least one of the application-specific policy controls.

Samsung Electronics Co., Ltd. (“Petitioner”/”Samsung”) petitions for IPR of claims 1-18 (“Challenged Claims”) of U.S. Patent 10,028,144 (“’144 Patent”).

I. IPR REQUIREMENTS

A. Standing

The ’144 Patent is available for IPR. This petition is being filed within one year of complaint being served against Samsung. Samsung is not barred/estopped from requesting this review.

B. Challenge, Relief Requested

Samsung requests an IPR of the Challenged Claims on the below grounds, which are supported by Dr. McDaniel’s testimony in SAMSUNG-1003 (¶¶1-279).

Ground	Claim(s)	35 U.S.C. § 102/103
1A	1-11, 13, 15, 18	§103: Wright
1B	12, 16, 17	§103: Wright-Tzannes
1C	14	§103: Wright-Smith

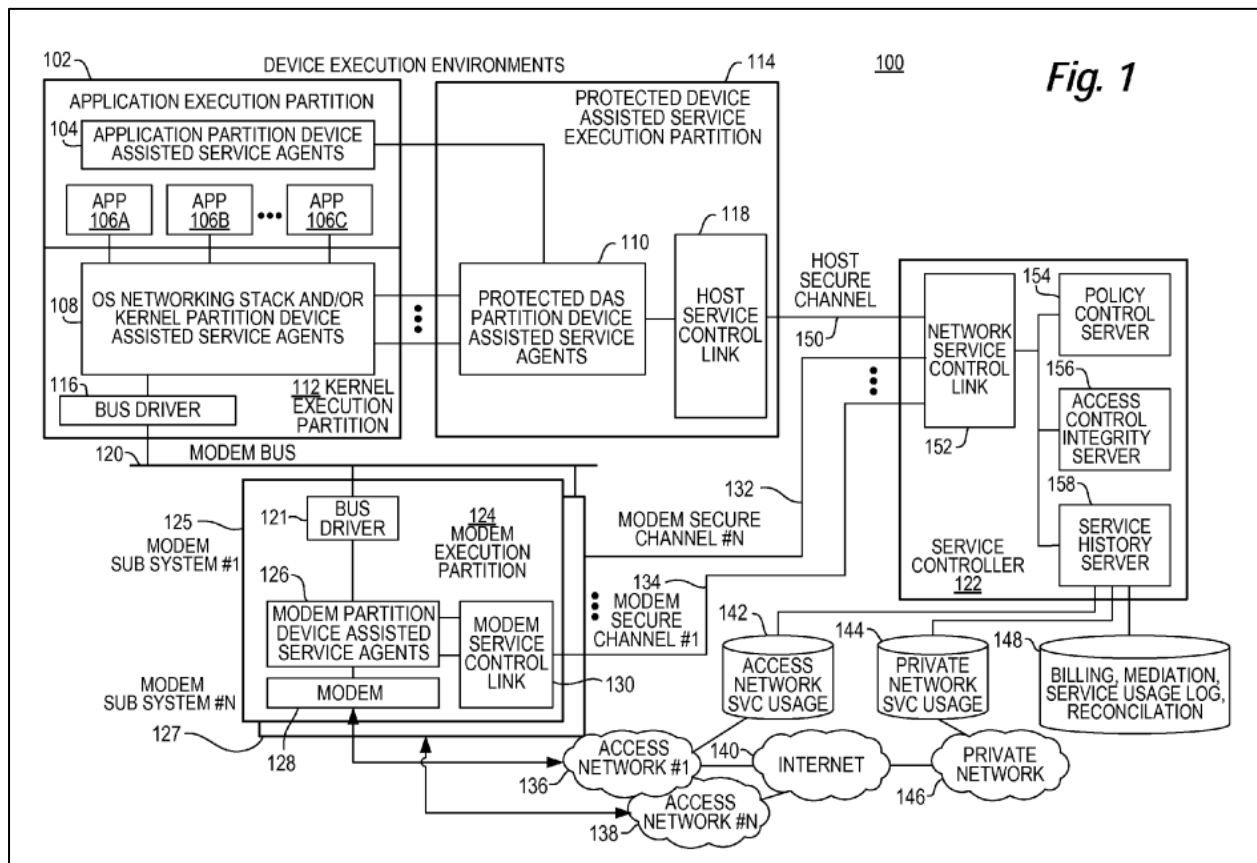
Reference	Filed	Published
Wright	10/29/2007	02/28/2008
Tzannes	4/12/2007	10/2/2008
Smith		2005 (SAM-SUNG-1012)

C. Claim Construction

No formal claim constructions are necessary here because “claim terms need only be construed... to resolve the controversy.” *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011); SAMSUNG-1003, ¶¶19-48. Petitioner reserves the right to respond to construction(s) offered/adopted in this proceeding. Petitioner is not conceding that claims satisfy all statutory requirements, nor waiving arguments/ground that cannot be raised here. Petitioner applies art consistent with Patent Owner’s infringement allegations. SAMSUNG-1035; SAMSUNG-1004.

II. ’144 PATENT

The ’144 Patent provides a mobile phone/device, which includes a wireless wide-area network (WWAN) modem and multiple partitioned execution environments (as shown below), and which receives policies and/or rules from a server. SAMSUNG-1001, FIG. 1, Abstract, 5:16-41. SAMSUNG-1003, ¶¶67-73.



The application for the '144 Patent was allowed without any rejections. SAM-SUNG-1002, 25.

III. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. GROUND 1A: Wright Renders Obvious Claims 1-11, 13, 15, and 18

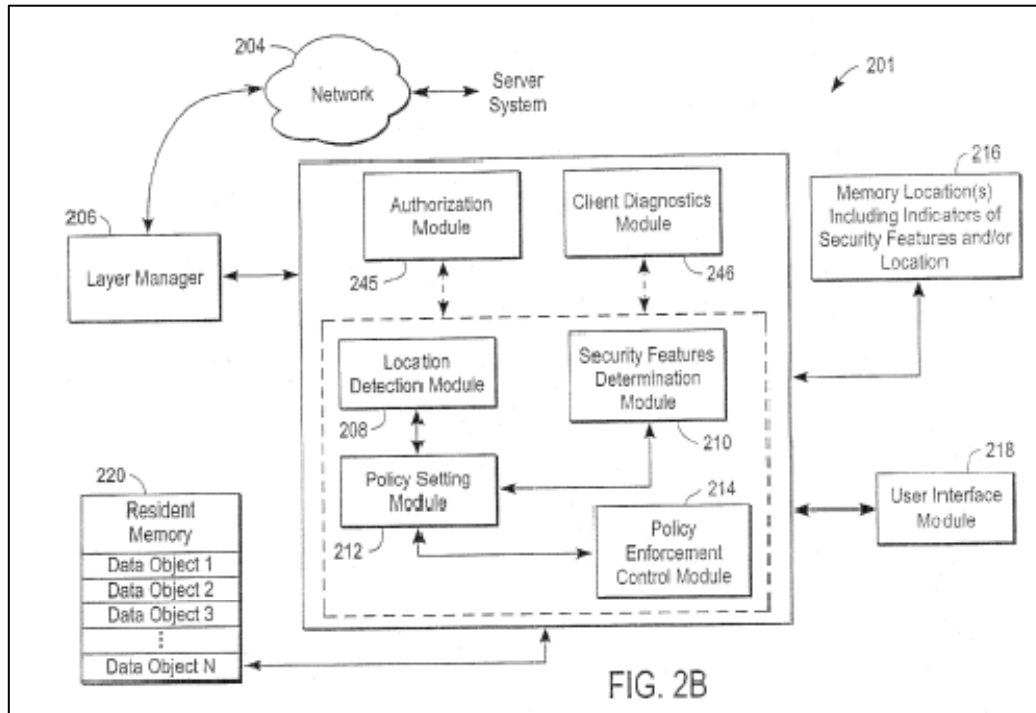
1. Wright

Wright provides a system for administering “protection of data accessible by” a mobile device. SAMSUNG-1005, ¶13; SAMSUNG-1003, ¶¶74-82.

Wright’s FIG. 2B (below) depicts a “mobile computer system” (referred as

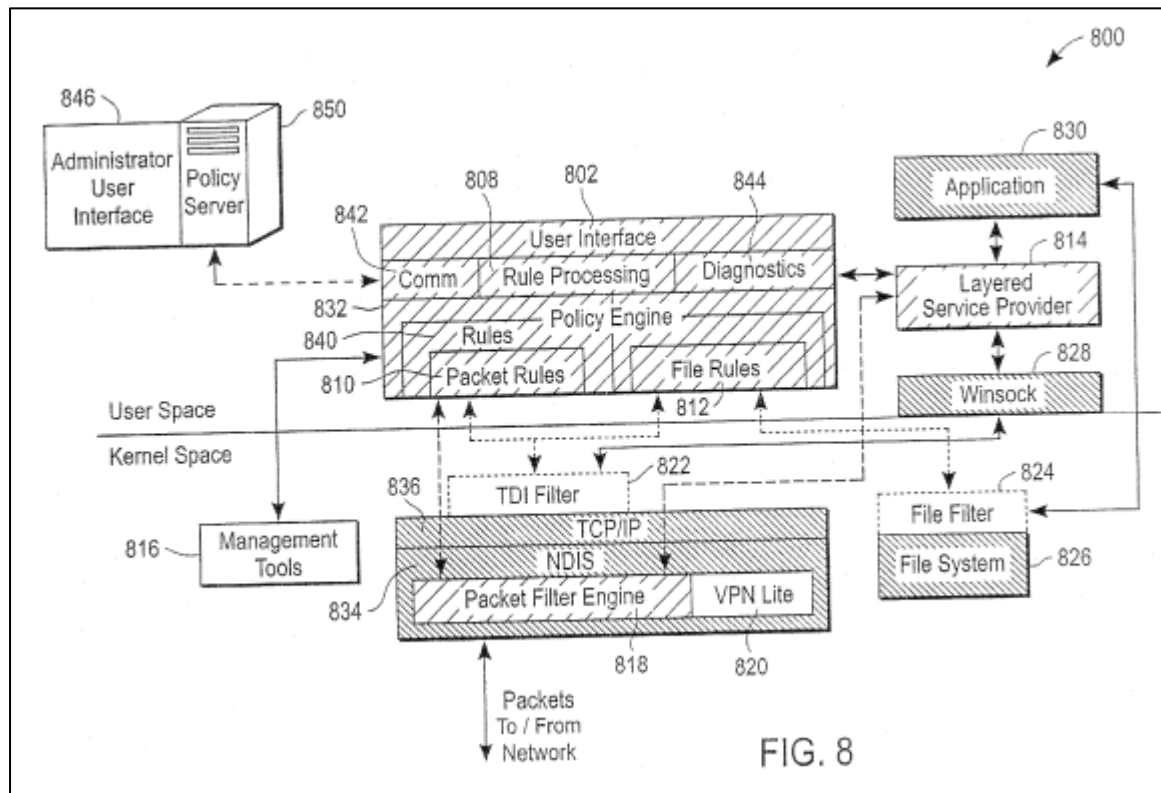
client system or client-side system 201) that communicates with a server system.

SAMSUNG-1005, ¶53.



As shown, client system 201 includes multiple components/modules for client-side management and enforcement of policies related to network traffic. SAMSUNG-1005, ¶¶65-67, 54. Client side system 201 “monitors the user’s changes in location and/or security features” (via modules 208 and 210) and “applies the appropriate policies automatically as the user moves... or [as] different security features” are activated/deactivated. SAMSUNG-1005, ¶¶53, 67-68. Policy setting module 212 “receives, installs and updates” security policies received from the server system, which are enforced by policy enforcement module 214. SAMSUNG-1005, ¶¶68, 71-73, 61, Abstract.

Like the FIG. 2 embodiment, Wright's FIG. 8 embodiment depicts and describes a server system (policy server) interacting with a client mobile device (over a network and via communication port 842) and enables protecting data accessible by the device. SAMSUNG-1005, ¶¶65, 235-237.



As shown, client side system is partitioned into a user/application space and a kernel space. SAMSUNG-1005, FIG. 8, ¶¶228, 237-238. The application/user space includes a policy engine 832, a user interface module 804, applications 830, and a layer services provider 814. *Id.*

Policy engine 832, which performs functionality similar to that of policy setting module of FIG. 2B, communicates with kernel components (e.g., components

824, 818, 822) that enforce policies and perform packet filtering. SAMSUNG-1005, ¶¶237-238, 283.

One or more applications 830 communicate, via an interface, with LSP 814, which “determine[s] which application... is accessing the network and what networking services the application will be using.” SAMSUNG-1005, ¶¶239-240. This information is used by the above-described kernel components, e.g., to perform enforcement of network traffic policies (and associated network packet filtering). SAMSUNG-1005, ¶¶283, 264-269, 239, 241 (filter determines application for which a packet is destined and decides whether to forward or filter traffic). SAMSUNG-1003, ¶¶58-59, 80-82.

2. Claim Analysis

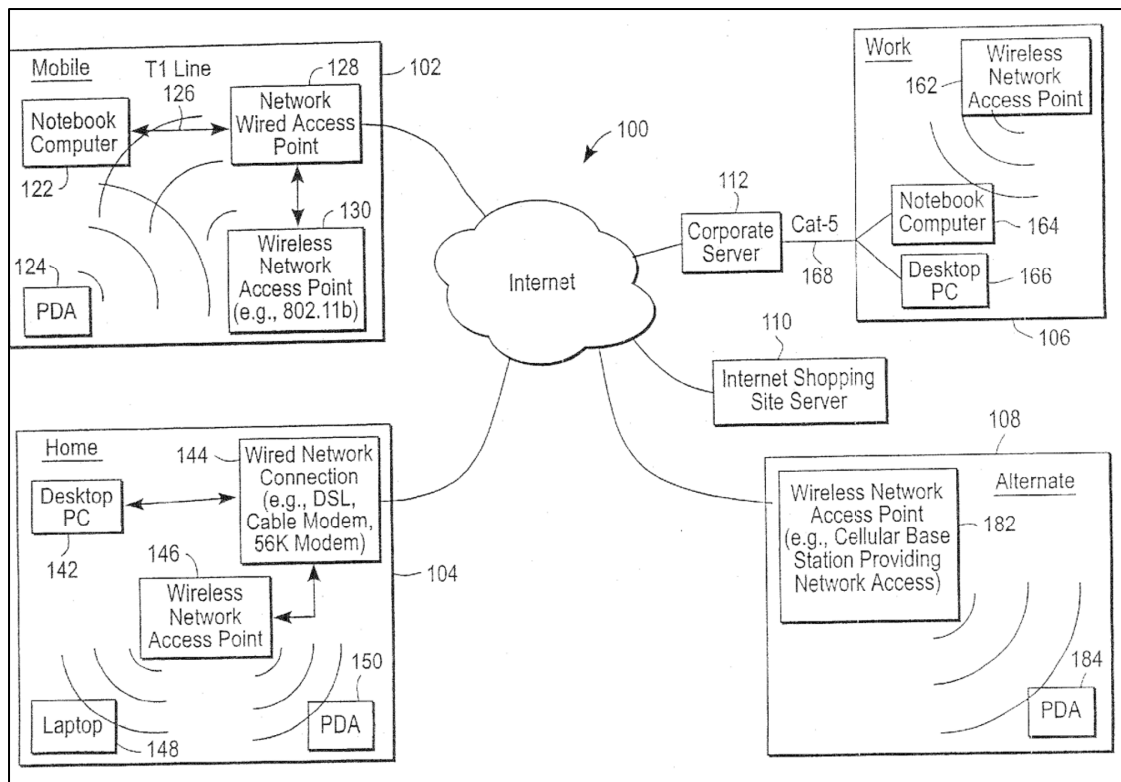
a) Claim 1

[1.1]

Wright discloses a *wireless end-user device*. Wright’s system and techniques “provide[] for the administration of the protection of data accessible by a mobile device,” which “include portable computing devices that access a network or another

computer.”¹ SAMSUNG-1005, ¶15. Per Wright, mobile devices are “wireless device[s]” because they can “access a network ...through a wireless connection.” *Id.*

As shown below, Wright teaches different types of end-user devices, *e.g.*, notebook computer 122, personal digital assistant (PDA) 124, notebook computer 164, PC Desktop 166, and mobile devices 148, 150, disposed at different locations across network 100. SAMSUNG-1005, ¶¶49-52, FIG. 1.



SAMSUNG-1005, FIG. 1

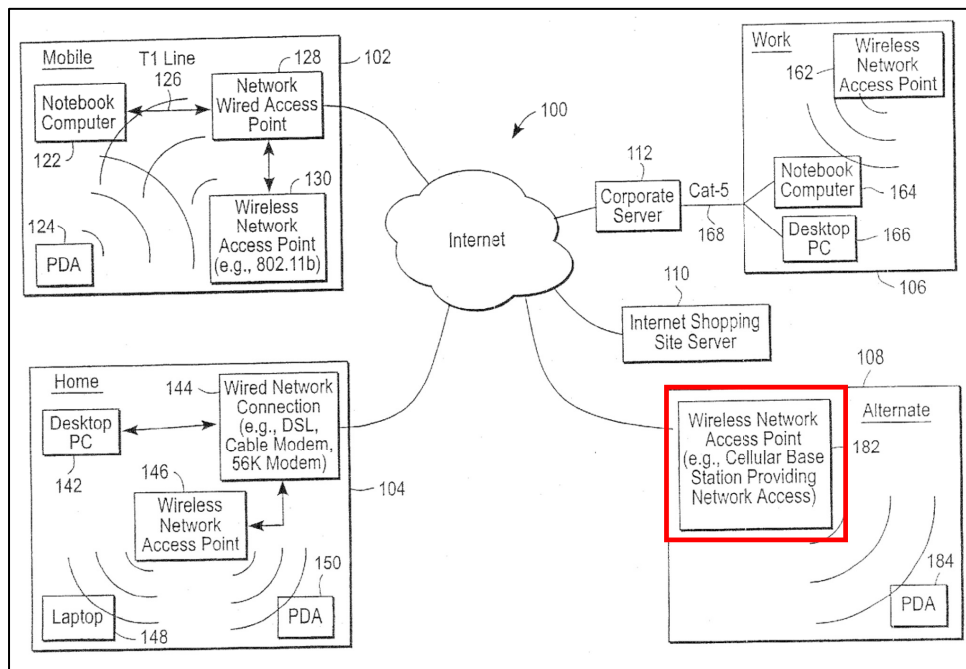
¹ Emphasis added throughout unless otherwise noted.

Similarly, in the context of FIGS. 2A-2B and FIGS. 7-8, Wright teaches systems implemented using mobile devices, which a POSITA² understood would include the example end-user devices described in the context of FIG. 1. SAMSUNG-1005, ¶¶53-84, 237-287; FIGS. 2A-2B, 7-8; SAMSUNG-1003, ¶¶110-114. Additional details of Wright's device are described below in [1.2]-[1.10].

[1.2]

Wright's end-user device communicates with wireless cellular networks, such as GPRS, GSM, and 3G. SAMSUNG-1005, ¶¶289, 293. Fig. 1 (below) shows and describes a mobile device (*e.g.*, PDA 184) communicating with a wireless network access point, which is “associated with a cellular base station providing network access through a” GPRS, GSM, 3G, or “other kind of mobile device wireless communication system.” SAMSUNG-1005, ¶¶52, 73, Fig. 1.

² See SAMSUNG-1003, ¶¶4-21, 24-26.



SAMSUNG-1005, FIG. 1 (annotated)

Network interface cards (NICs) or network adapters are implemented at Wright's device to facilitate device's wireless communication with such wireless cellular networks. SAMSUNG-1003, ¶¶113-118. Per Wright, "different security polic[ies]" are "applied for particular classes (e.g., 802.3, 802.11a, 802.11b, GPRS, GSM, 3G) of NICs" and/or policy engine determines "from the operating system[,] the class" (e.g., GPRS, GSM, 3G) of the network adapters on the mobile device. SAMSUNG-1005, ¶¶73, 289, 293.

A POSITA understood that (1) wireless cellular network (GPRS, GSM, 3G) are WWANs (SAMSUNG-1003, ¶116) and (2) a device's "network adapters"

(NAP) and/or “network interface cards” (NIC) that enable access to wireless networks are examples of modems. SAMSUNG-1003, ¶116; SAMSUNG-1013, 77-84. Because Wright’s device includes NICs and NAPs that enable access to one or more WWANs, a POSITA understood or found obvious that these components are **WWAN modems**. SAMSUNG-1003, ¶116; SAMSUNG-1013, 9; SAMSUNG-1023, 768. SAMSUNG-1045, 11:12-16.

Wright’s **WWAN modem receives and transmits Internet data between the device and at least one WWAN**. Per Wright, mobile device accesses the Internet via a WWAN (and its associated network access point). SAMSUNG-1003, ¶117. As FIG. 1 shows, a PDA communicates with a wireless network access point, e.g., “a cellular base station providing network access through a” GPRS, GSM, or 3G network, which in turn enables access to the Internet, such as e.g., to “an Internet shopping site server 110.” SAMSUNG-1005, ¶¶51-52, 85, 222, 231-232, 49-54; FIG. 1.

A POSITA understood or found obvious that Wright’s WWAN modem, which communicates, via a wireless cellular network, with servers or other devices on the Internet, would **receive and transmit Internet data between the device and at least one WWAN**. SAMSUNG-1003, ¶¶117-118. And such communications would happen when the **WWAN modem/device is configured for and connected to the at least one WWAN**, such as when the device/modem connects to the Internet

via a wireless network access point “associated with a cellular base station providing network access through” a wireless cellular network. SAMSUNG-1003, ¶118; SAMSUNG-1005, ¶¶51-52.

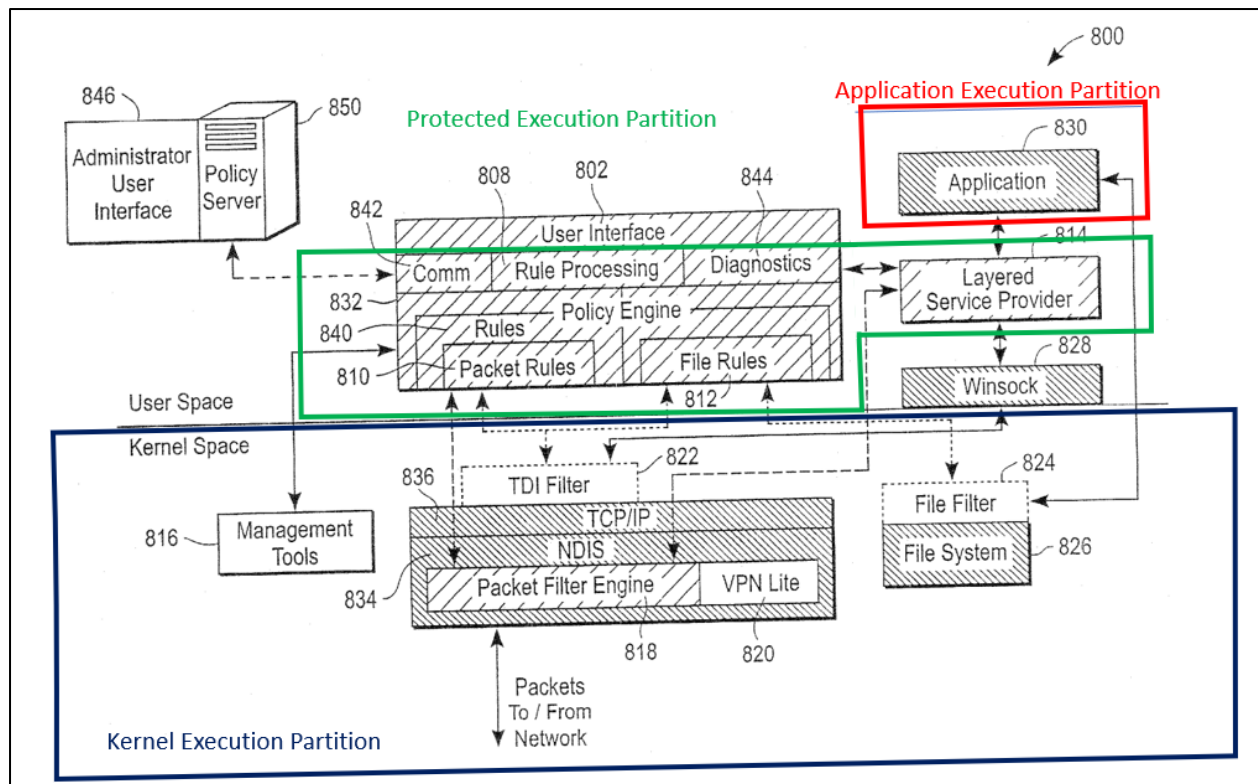
[1.3]

Wright’s *wireless end-user device* includes *one or more processors*. SAMSUNG-1005, ¶53; SAMSUNG-1003, ¶¶119-127. Wright’s disclosed functionality is “implemented as software executing in a mobile computing device having a processor” that is “accessible to memory” and is “communicatively coupled with one or more network interfaces....” SAMSUNG-1005, ¶¶53-84, 228-243, 286-292 (similar teaching for FIGS. 7-8 embodiments); SAMSUNG-1003, ¶119.

The device’s *one or more processors* are *configured to execute one or more processes in a kernel execution partition*. Wright’s device has a dedicated “kernel space,” which is shown in FIG. 8 as partitioned from the user space, thus confirming that its kernel space operates in a separate partition. SAMSUNG-1003, ¶120; *see id.*, ¶¶50-58. Wright’s kernel space includes several agents or processes executing therein (*e.g.*, driver 834, packet filter engine 818, TDI Filter 822, management tools 816, file filter 824, file system 826, VPN Lite 820) that are configured to perform one or more operations, *e.g.*, executing filters, updating parameters, granting/blocking network traffic. SAMSUNG-1005, ¶¶237-240, 243, 265, 281, 296. Thus,

Wright's components in the kernel space are *one or more processes executing in a kernel execution partition*. SAMSUNG-1003, ¶¶120-121.

Moreover, a POSITA understood that system 800 is “a mobile computing device having a processor,” which would be configured to execute instructions for performing one or more of the processes in the kernel space. SAMSUNG-1005, ¶¶53, 58, 63, 228-237; SAMSUNG-1003, ¶122.



SAMSUNG-1005, FIG. 8 (annotated)

Also, as shown above, Wright's device has a “user space” (referred to as “application space”) that includes policy engine 832, layered service provider 814, Winsock 828, and one or more applications 830. SAMSUNG-1005, FIG. 8. Per

Wright, policy setting module 212 (FIG. 2B) includes analogous functionality to that of policy engine 832. SAMSUNG-1005, ¶283. Thus, given Wright's teaching that a "stable execution environment" is "part of the policy setting module 212," a POSITA recognized such an execution environment would be provided by the similar "policy engine 832" and that this execution environment facilitates communication of "execution instructions" as policies exchanged between the client device and the policy server. SAMSUNG-1005, ¶¶286-287. Wright teaches secure (e.g., through use of encryption) receipt and management of policies received from the policy server and by policy engine 832 and LSP 814. SAMSUNG-1005, ¶¶274-283, 296.

Consistent with the '144 Patent's disclosure that a partitioned environment provides a secure execution environment (SAMSUNG-1001, 2:42-47), a POSITA understood or found obvious that the policy engine 832 and the layer services provider 814 are partitioned from other components in the user space (e.g., applications 830). SAMSUNG-1003, ¶¶123-124; 50-58. That's because Wright contemplates an execution environment implemented at the policy engine 832, and utilizes the policy engine 832 and layered services provider 814 for secure communication and management of policy information at the client device. SAMSUNG-1005, ¶137, 169, 22; SAMSUNG-1003, ¶¶123-124.

Additionally, it was well-known in the relevant timeframe to partition environments, particularly when doing so would enable securing sensitive data and functionality of a system. SAMSUNG-1003, ¶¶125, 50-58. Wright contemplates securely handling policies and ensuring proper system function (e.g., no corruption or tampering). SAMSUNG-1005, ¶138. Thus, applying well-known partitioning techniques to Wright's system would ensure secure and proper functioning of its policy storing and management operations. SAMSUNG-1003, ¶¶125-127, 49-55; SAMSUNG-1005, ¶¶137-138, 184.

Partitioning computing environments was a known and conventional technique (SAMSUNG-1003, ¶¶125, 50-58), and it would be predictable to combine such known techniques with known systems (Wright's system) to achieve predictable results of partitioned environments that are secured and separated from operations of other applications in the user space. SAMSUNG-1003, ¶¶56-57, 125; SAMSUNG-1014, 69; SAMSUNG-1022, 1283; SAMSUNG-1015, 41, 108. Additionally, a POSITA would have expected reasonable success in partitioning the user space, given that it was well-known and straightforward to partition environments and Wright's system already contemplates a type of partitioning with respect to its implementation of an execution environment for the above-contemplated secure policy handling and management. SAMSUNG-1003, ¶¶125-127, 50-58; SAMSUNG-

1005, ¶¶228, 237-238; SAMSUNG-1014, 69; SAMSUNG-1022, 1283; SAMSUNG-1015, 41, 108.

A POSITA therefore understood or found obvious that Wright's user space would be partitioned such that the (1) policy engine 832 and layer services manager would be in a secure/protected execution partition (*protected execution partition (PEP)*) and (2) applications 830 would be implemented in a separate partition (*application execution partition*). SAMSUNG-1003, ¶¶125-126, 50-58; SAMSUNG-1015, 41. Additionally, a POSITA understood or found obvious that user interface module would be implemented in the application execution partition, as explained in [18]. SAMSUNG-1003, ¶¶258-264; *see supra* [18].

Applications 830 are *one or more processes in an application execution partition*. SAMSUNG-1005, ¶¶187, 239; SAMSUNG-1003, ¶127; SAMSUNG-1040, 46; SAMSUNG-1041, 5:41-61; SAMSUNG-1025, 3, 16-17, 20, 64. A POSITA similarly understood that LSP 814 and policy engine 832 are processes executed by the processor that perform these components' operations and functionality (policy setting and management). SAMSUNG-1005, ¶¶237-243; SAMSUNG-1003, ¶127; SAMSUNG-1015, 27, 29, 41; SAMSUNG-1022, 1284.

[1.4]

Wright teaches or renders obvious that *Internet data packet traffic* passes between the *WWAN modem* and one or more applications executing in the *application execution partition*. SAMSUNG-1005, ¶¶73, 231-232, 281-282, 293. In Wright, an application can send data over a network, such as a file (e.g., email, text message, TIFF file, or a Word document) or can use a browser to access a website of an Internet shopping site. SAMSUNG-1005, ¶¶51, 73, 187.

Because Wright's applications (executing in the application execution partition) can send/receive data as network packets over the Internet (*supra* [1.2]), a POSITA understood that such transmissions happen from the respective applications via the device's NIC/NAP (*WWAN modem*, per [1.2] *supra*). *Id.*; SAMSUNG-1003, ¶¶128-134; SAMSUNG-1005, ¶282 (enabling a TLS/SSL socket for client-server communications to/from device applications).

Moreover, as noted *supra* [1.3], Wright's kernel space and associated processes correspond to *one or more processes executing in the kernel execution partition*. SAMSUNG-1003, ¶130.

Wright's kernel space includes *an operating system packet network stack*. Wright's mobile device "processes" incoming and outgoing "network traffic" using "information in one or more layers of a model for communications between computers in a network," such as the seven-layer OSI data communication model or the

four-layer Internet Protocol Stack. SAMSUNG-1005, ¶¶67, 220, 229. A POSITA understood that such multi-layer network communication models are network stacks (or are implemented as network stacks on computing devices, such as Wright’s mobile device). SAMSUNG-1003, ¶¶130-131; SAMSUNG-1016, 20-21; SAMSUNG-1016, 6.

Moreover, a POSITA understood or found obvious, in view of Wright’s teachings and the common knowledge of POSITA, that this network stack is typically implemented using drivers, components, and associated functionality in the kernel. SAMSUNG-1003, ¶132; SAMSUNG-1016, 6, 20-21. In Wright, a Network Driver Interface Specification (NDIS) driver “operat[es] within the kernel of the operating system of the mobile device” and enables network communications. SAMSUNG-1005, ¶¶228, 237, 239 (“network parameters” are obtained by “the NDIS driver for OSI layers 2-5”). Per Wright, “[n]etwork packets from” the NIC “are received by a Miniport Driver” and a transport driver determines information associated with the OSI model’s transport layer. SAMSUNG-1005, ¶229; SAMSUNG-1034, 8; SAMSUNG-1003, ¶132. Per Wright, the kernel’s filter engine 818 “process[es] parameters at the network layer in the OSI or IPS models” with respect to “network packets” received from the NIC. SAMSUNG-1005, ¶229.

A POSITA understood that the above-described Internet packet network traf-

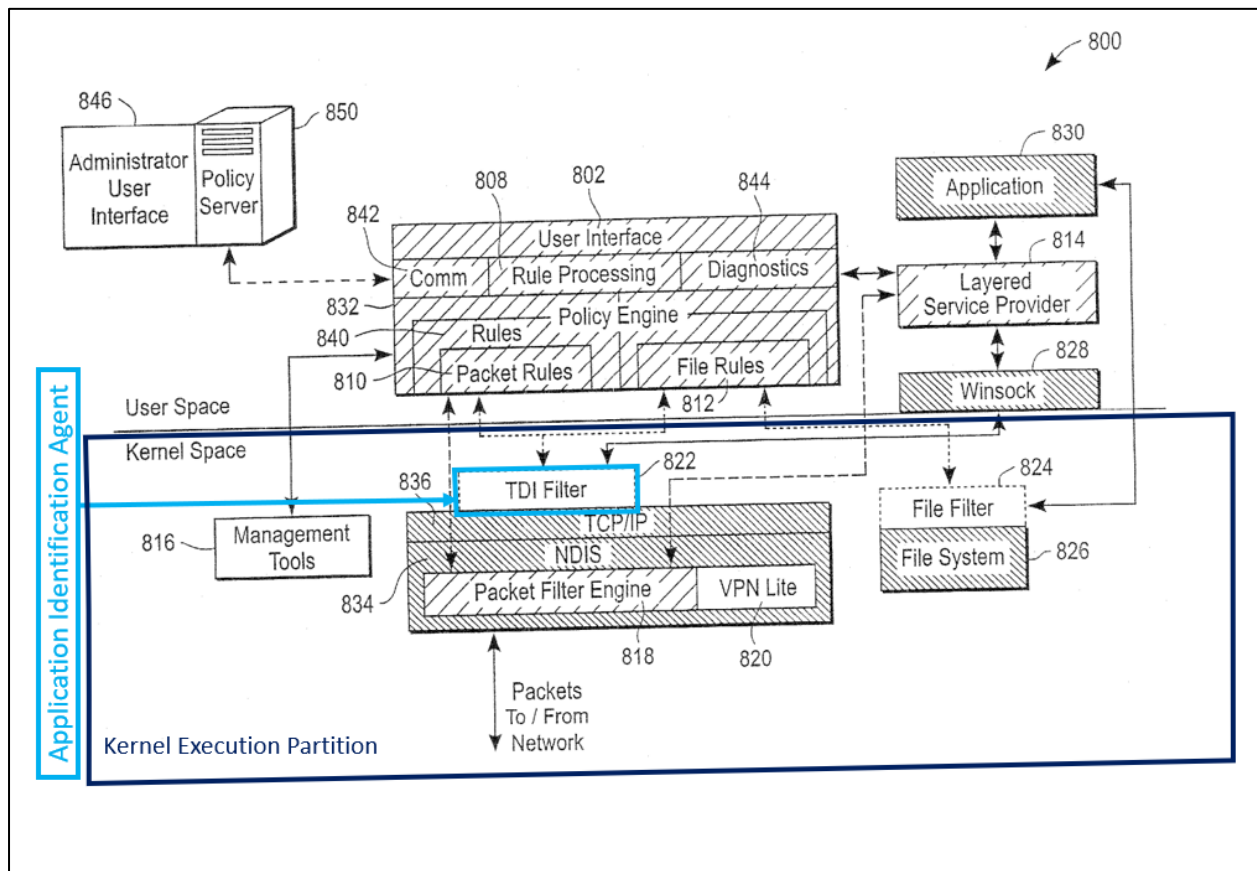
fic between the WWAN modem and the applications would pass through the *operating system packet network stack* to facilitate the network communications contemplated in Wright. SAMSUNG-1003, ¶¶133-134; SAMSUNG-1005, ¶¶228-229, 237; SAMSUNG-1016, 6.

A POSITA therefore understood that the *operating system packet network stack* (e.g., drivers, components, and associated functionality implementing the OSI/IPS network models) is configured *to pass Internet data packet traffic* (e.g., data related to “a file,” “an e-mail, a text message, a TIFF file, or a Word document”) *between the WWAN modem and one or more applications* (e.g., an application 830 handling such data) *executing in the application execution partition*. SAMSUNG-1005, ¶187; SAMSUNG-1003, ¶134.

[1.5]

Wright’s kernel space (*kernel execution partition*) is configured to execute *one or more processes*. *See supra*, [1.3]; SAMSUNG-1003, ¶¶135-140.

A POSITA understood that one of these processes, application/TDI filter 822 (shown below) corresponds to the *application identification agent* that *classifies individual flows of the Internet traffic through the stack according to which of the one or more applications is associated with a particular individual flow*, for the reasons described below. SAMSUNG-1003, ¶136.



SAMSUNG-1005, FIG. 8 (annotated)

Application/TDI filter 822 applies multiple classifications of packets/traffic passing therethrough. Per Wright, “NDIS filter driver” acts on low level information *i.e.* TCP or UDP port numbers” and “[w]hen the packet arrives at” TDI filter 822, it identifies applications because it “determines based on one or more application parameters for which... application a packet is destined...” SAMSUNG-1005, ¶241.

Filter 822 performs “application-specific filtering” on the individual packets associated with particular applications and based on application parameters. SAM-

SUNG-1003, ¶¶137-138; SAMSUNG-1005, *id.*, ¶¶264-269, 271, 283, 297 (TDI Filter 822 serves as an “enforcement mechanism[],” similar to Fig. 2’s enforcement control module 214”).

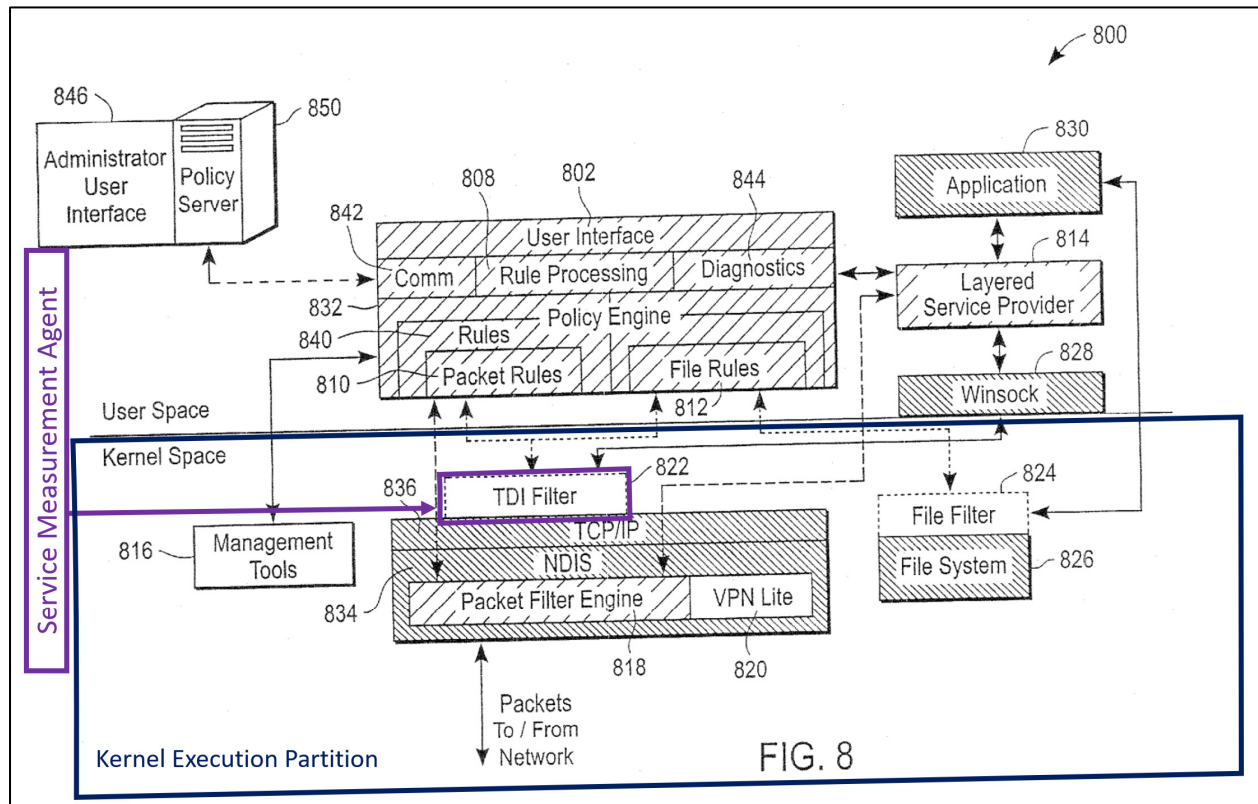
By determining which application corresponds to particular packet traffic and applying application-specific policy rules thereto, a POSITA understood or found obvious that TDI filter 822 is an *application identification agent* included *in the kernel execution partition* that *classifies individual flows* of packets or traffic *according to which of the one or more applications is associated with a particular individual flow*. SAMSUNG-1003, ¶¶137-139.

Moreover, the traffic flow relates to the Internet data traffic between the WWAN modem and the one or more applications, which a POSITA understood would implicate *traffic passing through the OS network stack* (which, per [1.3] *supra*, facilitates network communications). SAMSUNG-1003, ¶140; SAMSUNG-1005, ¶¶215, 229, 267, 283, 67, 237, 297. Thus, a POSITA would find obvious that Wright’s network packets or traffic that are/is application-specific *passes through the stack*. *See supra*, [1.4]; SAMSUNG-1003, ¶140.

[1.6]

As explained in [1.3] *supra*, TDI filter 822 and the packet filter engine 818 are processes executing in the kernel space (*kernel execution partition*).

These components perform “stateful filtering,” which can be performed “on a packet basis” and enables “refined application filtering” “based on application parameters.” SAMSUNG-1005, ¶¶265, 268, 297, 193, 199; SAMSUNG-1003, ¶141.



SAMSUNG-1005, FIG. 8 (annotated)

Wright contemplates “peer to peer” communication in which files are exchanged between devices, or downloaded from the server during a communication session. SAMSUNG-1005, ¶¶135-181; FIGS. 3A-3F. Per Wright, policies defined for such communications control the “type, size, format, and other attributes of transferred files.” SAMSUNG-1005, ¶169. Such file attribute-based policies are appli-

cable to, e.g., email applications and files (e.g., attachments) sent or received therefrom. *See id.*, ¶169; ¶¶158-163 (using “software update size” or “policy size” to calculate an estimated download time.”); SAMSUNG-1003, ¶¶59-65. A POSITA therefore understood that attributes of files being transferred by an application are application parameters because such information would be derived by the application layer of the network stack/OSI model (e.g., when an application indicates that it is sending such files over a network). SAMSUNG-1003, ¶¶143, 59-65; SAMSUNG-1005, ¶193 (“application parameter [is] typically associated with the application layer in the OSI or IPS models”).

Given the above disclosures, a POSITA understood or found obvious that Wright’s application-specific traffic filtering can be dictated by policies based on application parameters such as size of packet traffic (e.g., size of file being transmitted) to/from the application. SAMSUNG-1003, ¶¶144, 59-67.

For Wright’s system to perform packet filtering based on traffic size (e.g., size of a file or attachment to be sent over a network), a POSITA understood that the system would measure the file/traffic size, or would find obvious to do so. SAMSUNG-1003, ¶145. Indeed, in the relevant timeframe, it was well-known to perform application-specific packet filtering at least in part by measuring/monitoring network packet traffic to/from the application, and such packet traffic/file size measuring functionality was conventionally implemented as part of kernel processes, as

corroborated by Taekuchi and Oester. SAMSUNG-1003, ¶¶145-146, 59-67; SAMSUNG-1009, ¶69; SAMSUNG-1010, ¶¶1, 10, 18, 21, 30-35; SAMSUNG-1017, 1; SAMSUNG-1024, 2.

Thus, implementing a measuring unit to measure size of network traffic would be an obvious, conventional way to implement what Wright describes, and is nothing more than utilizing familiar, known techniques to achieve a predictable result of application-specific filtering based on packet size (per Wright). SAMSUNG-1003, ¶147. Alone or in combination, Wright's TDI filter 822 and filter engine 818 provide an obvious kernel implementation of measuring an amount of the Internet data packet traffic associated with each of Wright's one or more applications, e.g., applications 830. *Id.* A POSITA would find obvious to provide such measuring functionality as part of the component(s) that perform application filtering based on such measurements (which, in Wright, includes TDI filter 822 and packet filter engine 818). SAMSUNG-1003, ¶147; SAMSUNG-1017, Abstract, §§2.2, 3.4-3.6; SAMSUNG-1024, 2-5, 10-11. A POSITA therefore understood that measuring traffic at TDI filter 822 or packet filter engine 818 provided a more secure traffic/file size measurement point in the kernel space (e.g., as compared to user space) by components already tasked with enforcement of traffic size-based policies. SAMSUNG-1003, ¶147; SAMSUNG-1005, ¶¶239, 167. Moreover, a POSITA would find obvious to implement this functionality as part of the TDI Filter 822 / Packet Filter 818

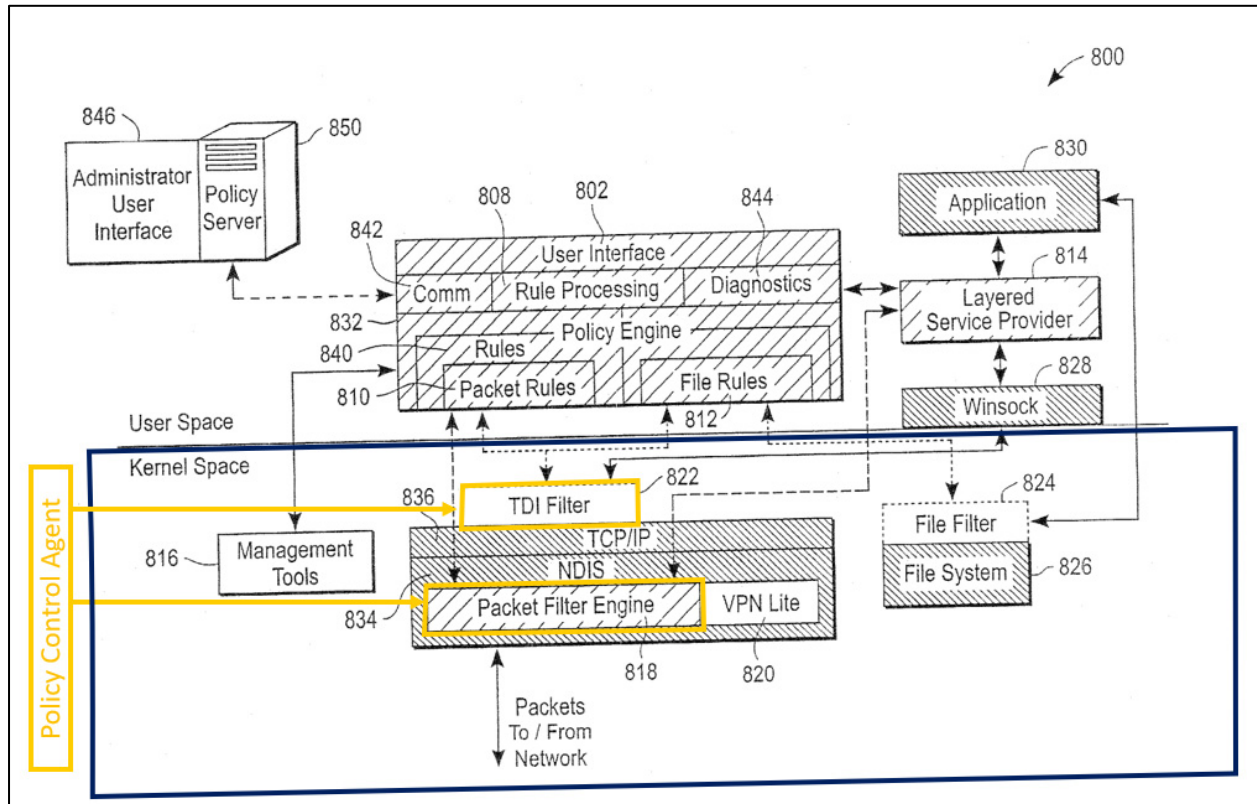
at least because (1) these are kernel processes (which is a conventional obvious place to implement such functionality, as explained above), (2) these processes are already tasked with application-specific filtering of network traffic based on traffic/packet size (SAMSUNG-1005, ¶¶242, 268-269; *see supra* [1.5], [1.8]), and (3) traffic size measurements would enable and further Wright's goal of more "refined" application filtering (SAMSUNG-1005, ¶268) by enabling granular filtering using measurements of size or volume of incoming/outgoing traffic. SAMSUNG-1003, ¶¶147-149, 59-65.

Thus, Wright in view of conventional knowledge of a POSITA teaches or renders obvious implementing *a service measurement agent* (TDI Filter and/or packet filter that would include packet size measurement functionality) executing in the kernel execution partition *that measures, for classified individual flows, an amount of the Internet data packet traffic associated with each of the one or more applications*. SAMSUNG-1003, ¶¶141-149.

[1.7]

A POSITA understood or found obvious that the TDI filter and packet filter engine in the kernel space (*kernel execution environment*) correspond to the claimed *policy control agent that appl[ies] application-specific traffic policy controls to classified individual flows*, as explained below.

As explained in [1.5], TDI filter 822 and packet engine 818 perform application-specific filtering using rules 840 from policy engine 832 (classifies individual flows of Internet packet traffic associated with applications). SAMSUNG-1003, ¶¶150-156; SAMSUNG-1005, ¶¶241, 269. In FIG. 8's embodiment, packet filter engine 818 and TDI filter 822 perform policy-based traffic filtering (e.g., application-specific traffic filtering) based on policy-driven rules 840. SAMSUNG-1005, ¶¶267-268; *see id.*, ¶¶263-266, 268-274, 297, 231; SAMSUNG-1003, ¶¶150-152.



SAMSUNG-1005, FIG. 8 (annotated)

In Wright, packets, which have been classified previously and filtered at the TDI filter 822 (*see supra* [1.5]), are forwarded to packet filter engine 818 to perform

“stateful packet filtering” that “filters packets based on a state set by the currently enforced policy with respect to a parameter.” SAMSUNG-1005, ¶¶265, 297. Additionally, rules, from the policy engine, are “passed to” TDI filter 822 for “application specific filtering.” SAMSUNG-1005, ¶269, FIG. 8; *see id.*, ¶297 (TDI filter filtering upper layers of network traffic for applications based on application parameters).

Given the above, a POSITA understood or found obvious that, TDI filter 822 applies application-specific rules for enforcement of network traffic policies (*applying application-specific traffic policy controls*) to classified individual flows of Internet data packet traffic associated with applications (*see supra* [1.5]). SAMSUNG-1003, ¶¶58-65, 153-155. Examples of Wright’s *application-specific traffic policy controls* include application-specific traffic rules or policies, e.g., relating to application-specific filtering or application-specific network access, and application-specific rules or policies as they relate to attributes of a file or object, such as size, type, etc. SAMSUNG-1005, ¶¶169, 238-239, 269; *see id.*, ¶¶55, 68, 72, 85-109, 158-163, 218.

Additionally, a POSITA understood or found obvious that packet engine 818, in NDIS driver 834, handles lower layer filtering/processing of network packets. SAMSUNG-1003, ¶156. Per Wright, location-based rules are used for traffic policy enforcement based on “network parameters obtained by the NDIS driver [which is part of packet engine 818] for OSI layers 2-5....” SAMSUNG-1005, ¶239. Per

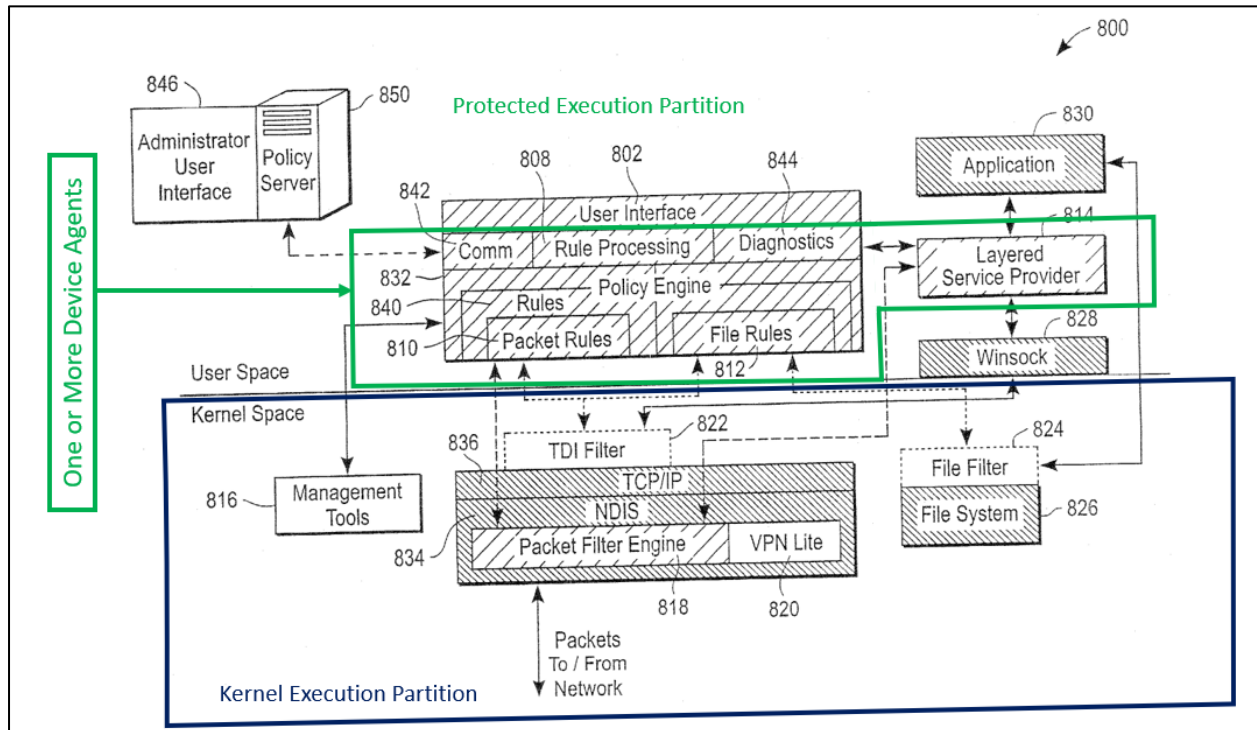
[1.5]-[1.6] *supra*, a POSITA understood or found obvious that TDI filter 822 classifies and measures individual flows of Internet data packet traffic. A POSITA therefore understood or found obvious that such classified/measured packets are provided for application-specific filtering (*applying application-specific policy controls to classified individual flows*) at packet engine filter 818. SAMSUNG-1003, ¶¶60-66, 155-156.

[1.8]

PEP components (LSP 814 and policy engine 832; *see supra* [1.3]) are *one or more device agents* because they are resident on Wright’s end-user device and are software programs configured to perform specific functions. SAMSUNG-1003, ¶¶157-163; SAMSUNG-1029, 3; SAMSUNG-1030, 3; SAMSUNG-1005, ¶¶237-301.

A POSITA understood or found obvious these *one or more device agents* have *privileges to access processes executing in the kernel execution partition* (kernel space). SAMSUNG-1003, ¶161. LSP 814 has privileges to access the packet filter engine in kernel space. SAMSUNG-1005, ¶281 (diverting incoming/outgoing application-specific traffic to kernel components), ¶238. LSP 814 also “captures information about network applications starting and stopping” and determines

ports/services “the applications will be using”—and subsequently passes this information to packet filter engine 818. SAMSUNG-1005, ¶239. LSP 814 has “a communication interface” with the packet filter engine 818 to facilitate the above described communications. SAMSUNG-1005, ¶237.



SAMSUNG-1005, FIG. 8 (annotated)

Additionally, policy engine 832 has “communication interface[s]” to processes in the kernel space, including (1) “management tools 816 of the operating system” that provide information about running applications/processes and types of adapters connected to the device, (2) “file filter 824 operating in the kernel space that controls access to the file system 826,” and (3) “packet filter engine 818 operating within a driver 834” to which rules are passed, from rules 840, to perform, e.g.,

stateful packet filtering. SAMSUNG-1005, ¶¶237, 243, 268-269, FIG. 8.

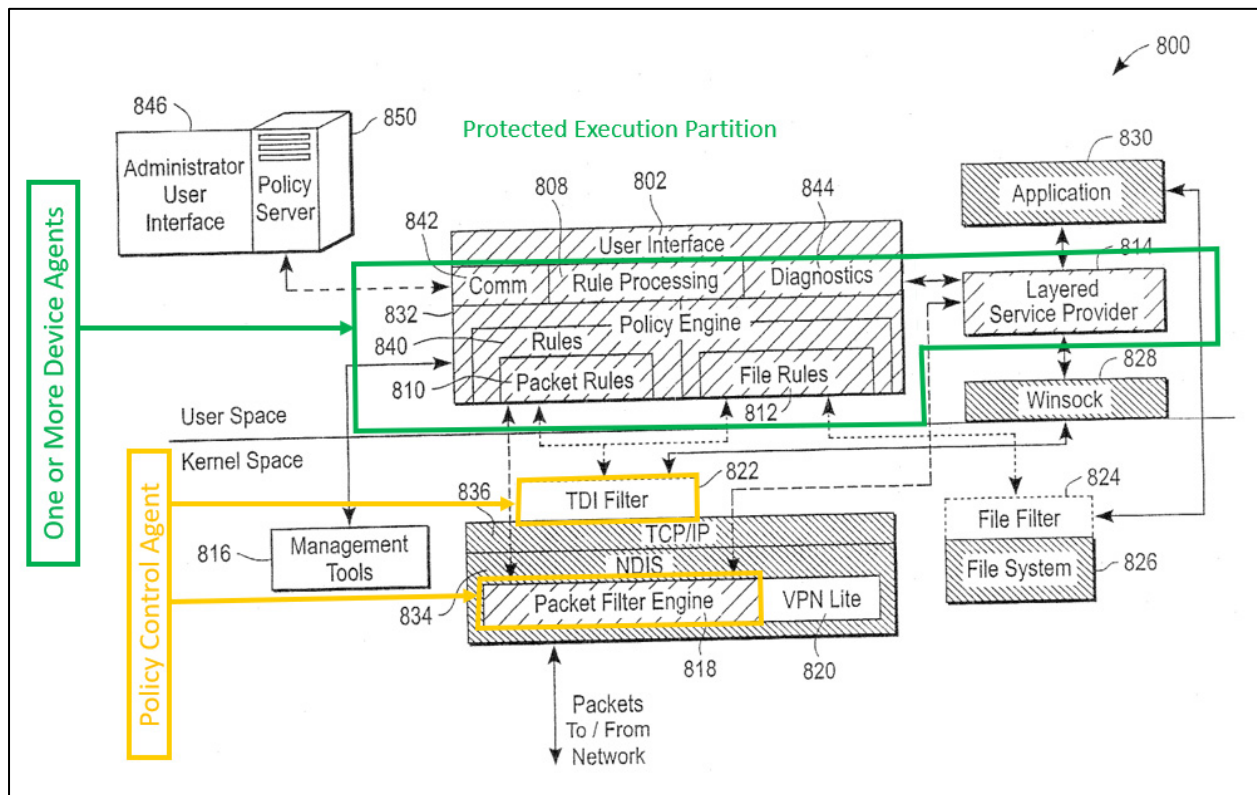
A POSITA understood or found obvious that the above-described *one or more device agents have limited privileges to access processes executing in the kernel execution environment*. SAMSUNG-1003, ¶¶161-163. A POSITA understood that the kernel is responsible for managing system resources, memory, and drivers for the mobile device, and that processes in the user space can only access processes executing in the kernel if they have permission to do so. SAMSUNG-1003, ¶161. Additionally, it is well-known, in the context of operating systems on computing devices, for processes in the user space (e.g., LSP 814, policy engine 832) to have limited privileges to processes in the kernel space, and that such limited privileges ensured security, stability, and proper operation of the overall system. SAMSUNG-1003, ¶¶161-162; SAMSUNG-1015, 17, 34, 26-27, 41, 79, 108; SAMSUNG-1014, 69; SAMSUNG-1022, 1283.

Thus, when considering the above and Wright's disclosure of only limited kernel processes with which LSP 814 and policy engine 832 communicate (as described above), a POSITA understood or found obvious that these components have *limited privileges to access processes executing in the kernel execution environment*. SAMSUNG-1003, ¶¶162-163.

[1.9]

As explained below, Wright’s policy engine 832 (*one or more device agents in the PEP*)—per [1.8] *supra*—has a privilege to configure *application-specific traffic policy controls* of *policy control agent* (packet filter engine, TDI filter). SAMSUNG-1003, ¶¶164-167. Per Wright, rules 840 of policy engine 832 “define one or more security policies” (SAMSUNG-1005, ¶245), and based on “rules 840” applied “to the current policy,” packet filter engine 818 and/or the TDI filter 822 perform application-specific packet filtering. SAMSUNG-1005, ¶¶268-269.

In FIG. 2’s embodiment, “policy setting module 212 receives, installs and updates the security information including security policies and/or software updates received from the policy management module 236 via the policy distribution module 234” over the network. SAMSUNG-1005, ¶68. Similarly, for FIG. 8’s embodiment, Wright teaches that policy engine 832—which “performs implementation examples of functions... of determining policies analogous to those of the policy setting module 212” (SAMSUNG-1005, ¶283)—“download[s]” policies “from the policy server 850.” SAMSUNG-1005, ¶¶244, 286-287. Given these disclosures, a POSITA would recognize that the functionality of the policy setting module would be implemented as part of the policy engine 832 and would be used to configure (*e.g.*, update and store) the policies in rules 840. SAMSUNG-1003, ¶¶164-166.



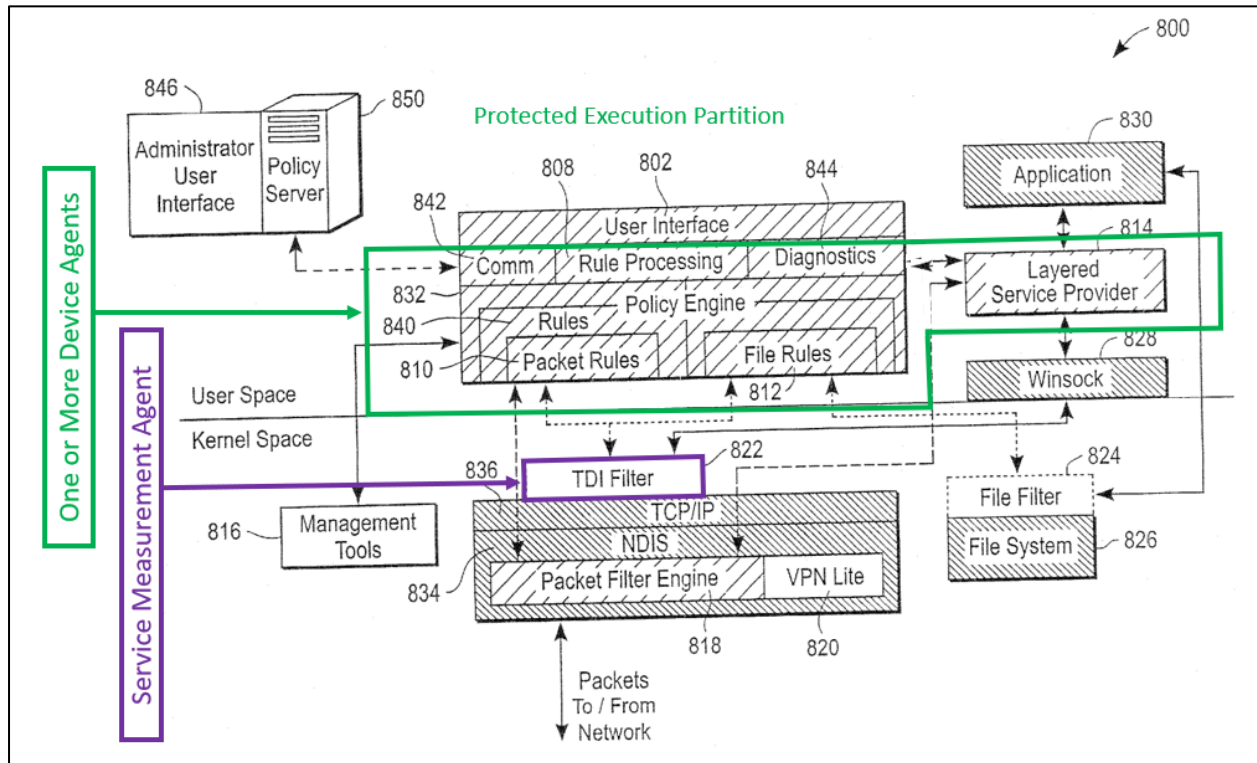
SAMSUNG-1005, FIG. 8 (annotated)

Thus, a POSITA understood or found obvious that Wright's policy engine 832 (*one or more device agents*) includes rules 840, which provides policies/rules used for application-specific filtering or network access (*a privilege to configure application-specific traffic policy controls*) performed by the packet filter engine 818 and TDI filter 822 (*policy control agent*). SAMSUNG-1003, ¶¶164-167; SAMSUNG-1005, ¶¶238, 245, 268.

[1.10]

Given [1.10] recites “a privilege to receive *at least one of* Internet data packet

traffic... **and** traffic information,” a POSITA understood that this limitation is satisfied by showing a privilege to receive either or both of the recited information. SAMSUNG-1003, ¶168. Even if both are determined to be required, Wright teaches or renders that obvious as explained below. SAMSUNG-1003, ¶¶168-174.



SAMSUNG-1005, FIG. 8 (annotated)

First, Wright teaches or renders obvious that the one or more device agents include *a privilege to receive at least one of Internet data packet traffic passing between the operating system packet network stack and the WWAN modem*. As explained above, Wright renders obvious that the *PEP* includes *one or more device agents* (LSP 814, policy engine 832) (*see supra* [1.8]), and that *Internet data packet*

traffic passes between the *operating system packet network stack and the WWAN modem* (*see supra* [1.4]).

LSP 814 has a *privilege to receive Internet data packet traffic*. Per Wright, LSP 814 has a communication interface with one or more applications 830, obtains “information about network applications starting and stopping and what ports the applications will be using,” and passes “this information to the packet filter engine 818, which then informs the policy engine 832 using an event signaling mechanism.” SAMSUNG-1005, ¶¶228, 239-240, 281 (diverting incoming/outgoing network traffic to other system components).

Wright further explains that LSP 814 provides “implementation examples of functionality analogous to that of layer manager 206” (SAMSUNG-1005, ¶283), which “process[es] network traffic” including “inbound data... accessible over a network and outbound copies of data objects from resident memory 220.” SAMSUNG-1005, ¶67. As explained above (*supra* [1.2]-[1.3]), such incoming and outgoing traffic corresponds to the *Internet data packet traffic* (e.g., packets, traffic, network traffic, data, and/or data objects 242), which would pass *between the operating system packet network stack and the WWAN modem*. Accordingly, Wright’s *one or more device agents* (LSP 814) has a privilege to receive *Internet data packet traffic passing between the operating system packet network stack and the WWAN modem*. SAMSUNG-1003, ¶¶169-171.

Additionally, or alternatively, Wright also renders obvious *a privilege to receive ... Internet data packet traffic* based on such traffic being received by policy engine 832 (*one or more device agents*). SAMSUNG-1003, ¶172. “Policy engine 832” downloads policy data from a policy over a network (*see supra* [1.9]), and diagnostics module 844 (part of policy engine 832), which performs similar functions [as] the client diagnostics module 246,” transmits, over a network, information such as “status, configuration, error logs, audit logs, and debug information to the server system.” SAMSUNG-1005, ¶¶284, 235-243, 61.

From these disclosures, a POSITA understood that policy data and diagnostics-related data received over a network is *Internet data packet traffic*, and such network traffic would pass through the operating system network stack and the WWAN modem (which facilitate network communications to/from the device). SAMSUNG-1026, 20; SAMSUNG-1003, ¶173. Thus, Wright’s policy engine 832 has *a privilege to receive Internet data packet traffic passing between the operating system network stack and the WWAN modem*. SAMSUNG-1003, ¶173.

Second, Wright teaches or renders obvious that the one or more device agents include *a privilege to receive traffic information including the measured amount of the Internet data packet traffic associated with each of the one or more applications*. In Wright, “policy engine 832, in accordance with the current security policy, provides” application-specific rules to” LSP 814, which “enforce[s]” such rules.

SAMSUNG-1005, ¶239. Because rules specific to applications can include size of files transmitted by/to applications (*traffic information including measured amount of Internet data packet traffic*) and LSP 814 enforces such a size-based rule, a POSITA understood or found obvious the measured amount (*traffic information*) would be received by LSP 814 to facilitate such policy enforcement. SAMSUNG-1003, ¶174.

b) Claim 2

Wright's device includes *one or more processors configured to execute* processes of *an application execution partition*, including application(s) 830. *See supra* [1.3].

LSP 814 (*at least one device agent executed in the protected execution partition*) communicates with applications 830 via a communication interface. SAMSUNG-1005, ¶¶238, ¶68 (“a communication interface, [can be] a software interface or buffers for storing parameters”), 58, 69. Via this interface, LSP 814 “determine[s] which application... accessing the network and what networking services the application will be using” and provides this information to other system components for application-specific policy enforcement. SAMSUNG-1005, ¶¶240-241; SAMSUNG-1003, ¶¶175-176.

A POSITA would recognize that application-specific policy enforcement is a

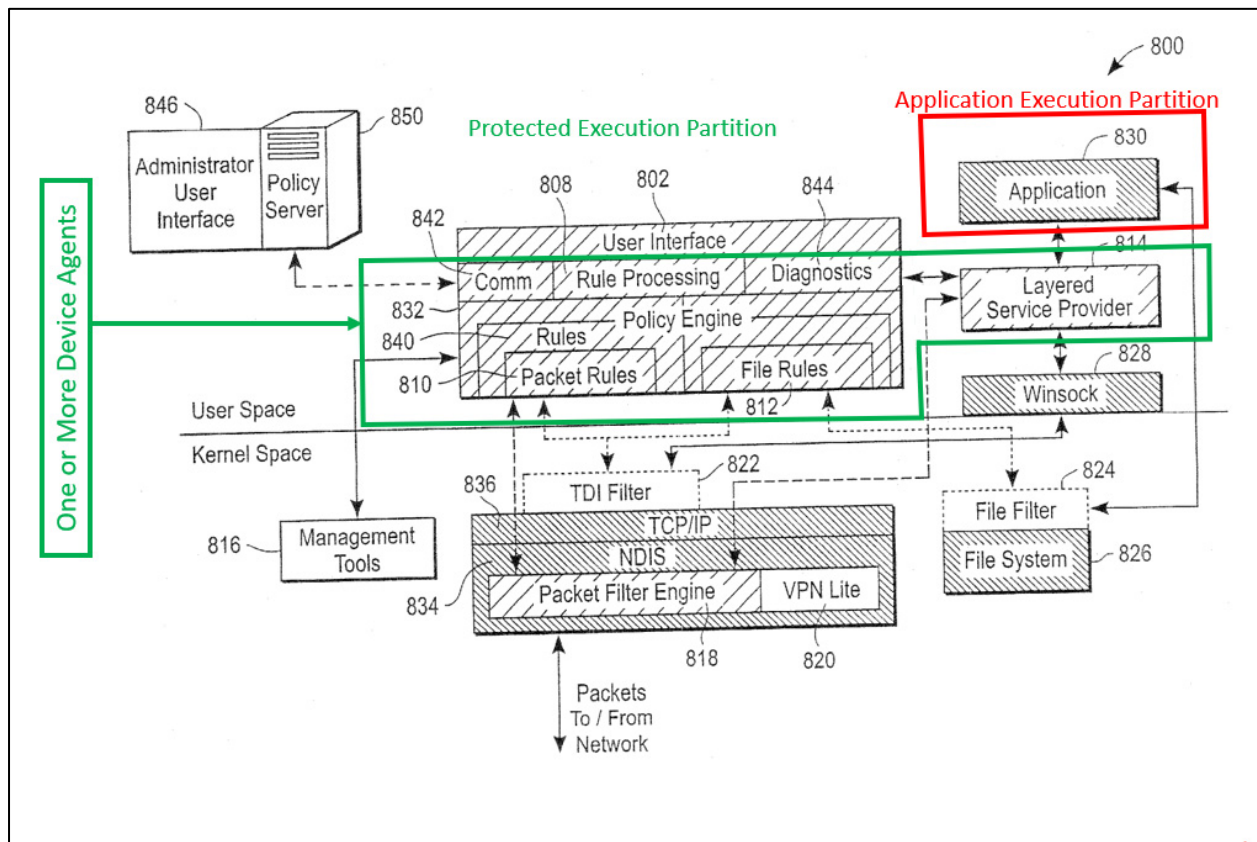
service provided by the device's agents/components. SAMSUNG-1003, ¶¶175-179. And, because the application-specific information is provided via communication interface to LSP for performing this service (policy enforcement), a POSITA understood that this interface assists in providing the policy enforcement device services and is therefore a *device assisted services agent*. SAMSUNG-1003, ¶175 (application interfaces are types of software agents tasked with facilitating inter-application communications); SAMSUNG-1015, 153.

Moreover, a POSITA understood or found obvious that this communication interface would be *executed within the application execution partition*. SAMSUNG-1003, ¶175. Per Wright, a communication interface can be implemented in different ways, e.g.: (1) "a bus between a processor executing one or more of the modules and a memory controller responsible for memory reads/writes," (2) "one module reading a parameter stored in a memory location by another module," and (3) "other communication interfaces known to" a POSITA. SAMSUNG-1005, ¶¶58, 68-69, 228; SAMSUNG-1003, ¶176. Indeed, applications were known to communicate using application programming interfaces (APIs). SAMSUNG-1003, ¶176; SAMSUNG-1018, 36; SAMSUNG-1015, 153. Implementing an API as part of or in relation to an application would have amounted to a known technique being applied to a known system (per Wright) to achieve predictable results of enabling Wright's application-to-LSP 814 communications. SAMSUNG-1003, ¶176.

Such an implementation is further motivated by Wright, which contemplates use of “known” communication interfaces (SAMSUNG-1005, ¶¶58, 68, 268), and would be straightforward and readily implemented by a POSITA, who would recognize APIs implemented by an application as a common and conventional communication interface for enabling application communications. SAMSUNG-1005, ¶169; SAMSUNG-1003, ¶177.

Wright also teaches that the “communication interface” can be “a software interface or buffers for storing parameters,” as received by LSP 814. SAMSUNG-1005, ¶¶68, 268, 228, 285, 187. Wright contemplating local storage of application-specific parameters would motivate a POSITA to implement the communication interface (e.g., APIs for the applications) at the respective applications or in an environment localized to applications (e.g., application execution partition).

A POSITA therefore found obvious to implement the above-described communication interface (*device assisted services agent*) in Wright’s *application execution partition*. SAMSUNG-1003, ¶¶178-179.



SAMSUNG-1005, FIG. 8 (annotated)

c) Claim 3

Wright renders obvious the recited processes (*respective processes*) executing on *one or more processors*. See *supra* [1.3], [1.5]-[1.7]; SAMSUNG-1005, ¶¶73, 228, 302-303; FIGS 1, 8; SAMSUNG-1003, ¶¶180-185.

These processes correspond to computer programs including programming instructions (*program code*). In Wright, (1) TDI filter 822 and packet filter engine 818 perform functions similar to policy enforcement module 214 (of FIG. 2), and (2) policy engine 832 performs functions similar to FIG. 2 modules 208, 210, 212.

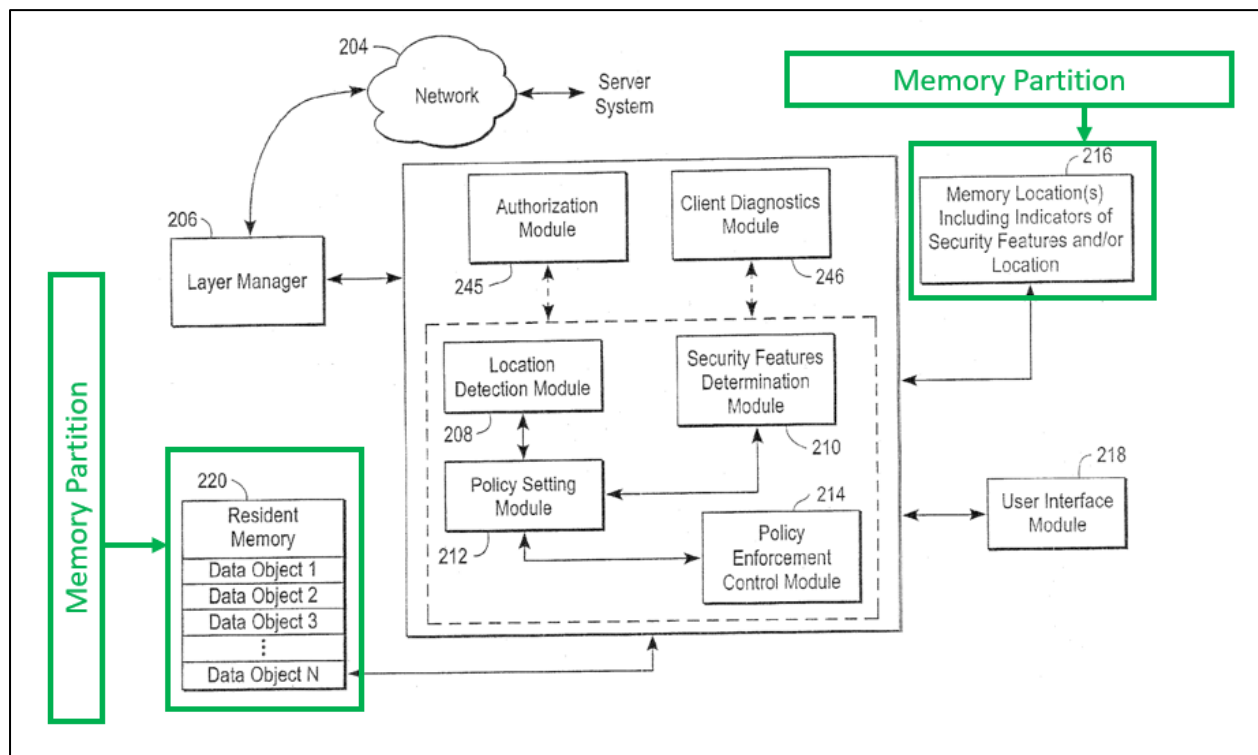
SAMSUNG-1005, ¶¶283, 301-303. These modules are “implemented as software” or within a “program” (*id.*) and are “embodied as instructions stored in one or more computer usable mediums some examples of which are a memory” (SAMSUNG-1005, ¶¶13, 65).

Given these disclosures, a POSITA understood or found obvious that Wright’s processes would be implemented as computer programs stored in its device memory (device *store[s] program code for the respective processes*). SAMSUNG-1003, ¶182.

Wright teaches or renders obvious *one or more memory partitions* for storing the program code for the above-described processes. SAMSUNG-1003, ¶183. Wright’s device includes memory (*e.g.*, memory locations 216 and resident memory 220), and as explained above, program code corresponding to the above-described processes would be stored in device memory. SAMSUNG-1005, ¶¶73, 228, 302-303; SAMSUNG-1003, ¶183.

Additionally, Wright’s device would include separate partitioned environments, with the processes in the kernel space being partitioned from those in the user space. SAMSUNG-1003, ¶¶184-185; *see supra* [1.3]. One obvious and conventional way to achieve such partitioning would be to partition the memory allocated to each partition and its underlying programs. SAMSUNG-1003, ¶¶183, 56-58.

Such memory partitioning was well-known in the relevant timeframe (*id.*) and leveraging such memory partitioning would amount to applying known techniques to known system to achieve predictable results (partitioned environment as rendered obvious by Wright). *Id.*; SAMSUNG-1014, 69; SAMSUNG-1022, 1283; SAMSUNG-1015, 41, 108; SAMSUNG-1015, 27.



SAMSUNG-1005, FIG. 8 (annotated)

A POSITA therefore understood or found obvious that the *program code* for the above-described processes (*respective processes*) would be stored in *one or more memory partitions*. SAMSUNG-1003, ¶184.

Moreover, per [1.3] *supra*, Wright discloses one or more processors to execute

processes of the above-described partitions. SAMSUNG-1003, ¶185. A POSITA understood that a processor would execute these processes by retrieving and executing the underlying program code from where it is stored in memory. SAMSUNG-1003, ¶185. A POSITA therefore understood that Wright's *stored program code* for the above-described modules/process would be retrieved from the *one or more memory partitions to execute the respective processes*. SAMSUNG-1003, ¶¶180-185.

d) Claim 4

[4.1]

As explained above for claim 2, Wright renders obvious *one or more memory partitions*, one of which would *store the program code for the one or more device agents*. SAMSUNG-1003, ¶186. This memory partition would be a *device-assisted services partition* because it stores program code for facilitating “service usage policy implementation” (given the policy control and management functions of LSP 814 and policy engine 832), which the '144 Patent explains is a device-assisted service. SAMSUNG-1001, 3:11; SAMSUNG-1003, ¶186.

A POSITA understood or found obvious that this partition is *separately protected*, e.g., from other memory partitions. SAMSUNG-1003, ¶187. It was conventional and well-known that memory partitions to be secured. SAMSUNG-1003,

¶187; SAMSUNG-1011, ¶¶17-36; FIGS. 2-3. Given that the device agents are programs for policy control and management, providing security over the corresponding portion of memory would beneficially avoid tampering with and ensuring proper operation of the device's policy-related functionality. SAMSUNG-1003, ¶187. There would also be a reasonable expectation of success in securing this memory partition, given the prevalent use of security mechanisms for memory partitions (e.g., encrypting information stored therein and/or limiting access). SAMSUNG-1003, ¶, 56-57, 187; SAMSUNG-1015, 17, 34. Thus, securing this memory partition would be a known technique to improve known systems (per Wright) to achieve predictable results (secure memory partitions storing program code). SAMSUNG-1003, ¶187. Wright's device-assisted services memory partition would therefore be separately secured, e.g., from other memory partitions that are commonly secured (e.g., memory partitions dedicated to kernel processes). SAMSUNG-1003, ¶187; SAMSUNG-1015, 17; SAMSUNG-1014, 69.

[4.2]

Wright's policy engine 832 (*one or more device agents*) includes a diagnostic module 844 and policy-setting and management functionality (providing similar functionality as policy setting module 212 and diagnostic module 246 of FIG. 2). SAMSUNG-1005, ¶¶283-284.

Wright's diagnostic module is an *access control integrity agent* because it observes the operations and service usage of device components and provides this to a server over the network for evaluation, which the '144 explains as "access control integrity" functions. SAMSUNG-1003, ¶¶188-190; SAMSUNG-1001, 6: 25-46; 9:55-65. System components 832, 814, 824, and 818 report incorrect or faulty operation to diagnostics module 844. SAMSUNG-1005, ¶270. This information is relayed to policy server 850 "as it indicates a possible compromise of the protective system" in which case "policy server 850 ...provides diagnostic support information to the diagnostics module 844" to remedy these issues. *Id.* The "support information may be in the form of instructions or code to the client device to repair a problem or perform maintenance," which would be implemented by the diagnostics module since "no user intervention or action" is required to perform such "repair" or "maintenance." *Id.*, ¶62; SAMSUNG-1003, ¶¶189-190.

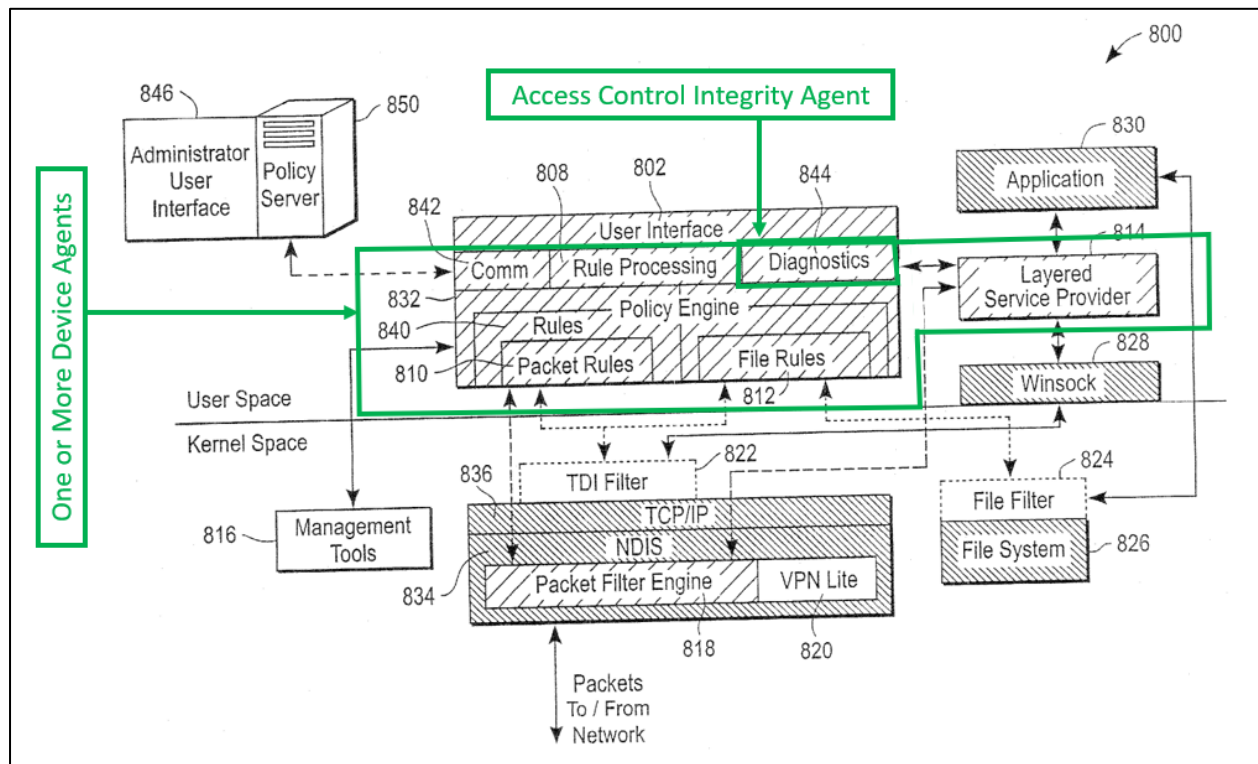
Moreover, diagnostic module would have *exclusive permission to update program code stored in the separately protected device-assisted services partition*. SAMSUNG-1003, ¶¶188-191. As explained above, the diagnostic module receives and automatically applies "code" to "repair the problem" or "perform maintenance," which a POSITA understood as updating the stored program code with the received code to remedy the issue. SAMSUNG-1003, ¶191. Because the received "code" update may be for LSP 814 (*device agent*) for which *program code is stored in the*

device assisted services memory partition (see *supra* [3]), a POSITA understood or found obvious that the diagnostic module has permission to update the program code for LSP 814 that is stored in this memory partition. SAMSUNG-1003, ¶¶188-191. Moreover, Wright only teaches that diagnostic module performs this operation, and conversely, no other module in Wright's system is described as providing the above function. A POSITA therefore understood that the diagnostic module 844 had an *exclusive permission to update* the above-described program code. SAMSUNG-1003, ¶191; SAMSUNG-1015, 26, 41, 79, 108; SAMSUNG-1022, 1283; SAMSUNG-1014, 69.

e) Claim 5

As explained below, Wright's *access control integrity agent* (client diagnostics module 246/diagnostic module 844; see *supra* [4]) is *further configured to validate the program code for the one or more device agents* (LSP 814, policy engine 832). SAMSUNG-1003, ¶¶192-193. In Wright, "policy engine 832, the layered service provider 814, the file filter 824, and the packet filter engine 818 report whether any of the other services have been disabled or removed from the system to the diagnostics module 844." SAMSUNG-1005, ¶270. Additionally, the diagnostic module "provides status, configuration, error logs, audit logs, and debug information to the server system." See *supra* [1.10]; SAMSUNG-1005, ¶¶284, 179, 270 (checks

for malicious software and other health checks), ¶83 (tests with respect to security policies). “This information [from the diagnostic module 246/844] may be relayed to the policy server 850 as it indicates a possible compromise of the protective system.” SAMSUNG-1005, ¶270; SAMSUNG-1003, ¶194. And, policy server 850 in turn provides, to the diagnostic module, “support information” e.g., “in the form of instructions or code to the client device to repair a problem or perform maintenance.” SAMSUNG-1005, ¶¶270, 62, 77-80.



SAMSUNG-1005, FIG. 8 (annotated)

From these disclosures, a POSITA understood or found obvious that the diagnostic module evaluates the operation of components such as the LSP 814 (*one or*

more device agents), and this evaluation of proper operation corresponds to validating program code for the corresponding functionality of this device agent. SAMSUNG-1003, ¶¶192-198.

Additionally, or alternatively, the diagnostic module would also be configured *to validate the program code for the one or more device agents* in view of Wright's teaching of security information (e.g., policy, software, or message) received by the policy engine 832 being authenticated, e.g., by the authorization module 245. SAMSUNG-1003, ¶196; SAMSUNG-1005, ¶¶137-138, 169, 247-252, 285. Since the received security information is authenticated/validated and this information includes security/policy-related code, this received and validated program code would be for policy engine 832, which would be understood as responsible for "install[ing]" and "update[ing]" policy related code and information. SAMSUNG-1003, ¶197; SAMSUNG-1005, ¶¶68, 283.

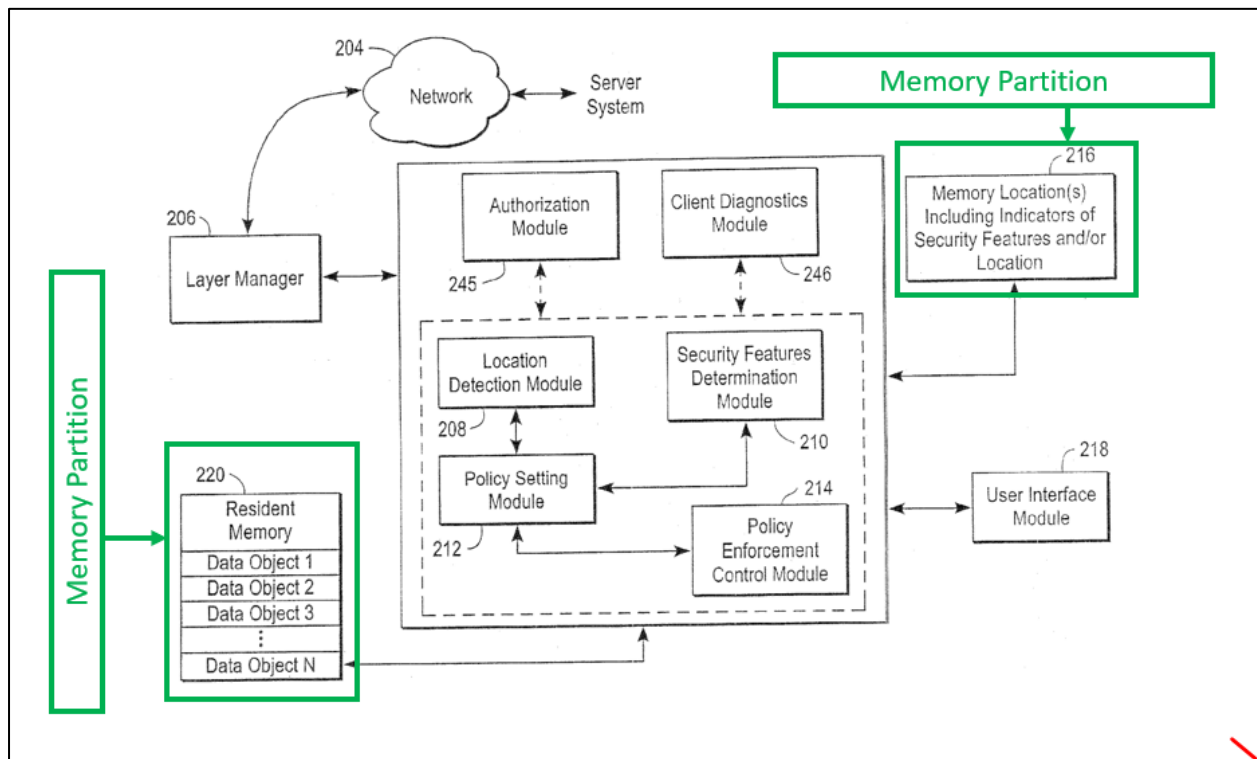
Additionally, per Wright, authorization module 245 performs the above-described validation of information received from the server system 200 (which is akin to the policy server 850, given its roles of disseminating policies). SAMSUNG-1005, ¶138; SAMSUNG-1003, ¶198. Although the authorization module is not depicted in the FIG. 8 embodiment, a POSITA understood this module to be implemented as part of the diagnostics module of FIG. 8. SAMSUNG-1003, ¶198. That's because both modules perform similar functionality—e.g., they both communicate

with the policy server (SAMSUNG-1005, ¶¶66-67, 270, 284) and both perform validations of code or other system functionality (SAMSUNG-1005, ¶¶62, 66, 84, 270, 284). Given that policy and other security information is conveyed to the policy engine in FIG. 8 and to the corresponding elements in FIG. 2B (e.g., policy setting module), and that diagnostic module of FIG. 8 is described as applying received code (similar to receipt and application of policy-related security information), a POSITA would have been motivated and would have reasonably expected success to implement the authorization module's functionality as part of the diagnostic module in FIG. 8's embodiment. SAMSUNG-1003, ¶198. Indeed, as noted previously, Wright's system aims to reduce tampering or modification of information exchanged between the device and the policy server (SAMSUNG-1005, ¶138) and policy engine 832 is responsible for managing policy-related information on the client device is policy engine 832. It would have therefore been obvious that the diagnostic module (*access control integrity agent*) is configured to *validate the program code for the one or more device agents*.

f) Claim 6

Wright teaches or renders obvious storing *device service state information*, consistent with how the '144 Patent uses this term. SAMSUNG-1003, ¶199; SAMSUNG-1001, 17:25-58.

As explained below, Wright teaches storing *device service state information* including diagnostic information, device configuration information, parameters, identifiers, information, reports, security features, policies, health checks or associated data and objects. SAMSUNG-1003, ¶200. As shown below, Wright's FIG. 2B shows on-device memory locations 216 and 220 that store "indicators of security features and/or location" and data objects. SAMSUNG-1005, FIG. 2B, ¶¶54, 184, 69. Per Wright, such data stored in memory includes "security policies, corporate data, group, files indicating the organization of personnel into various groups, client device management data, and diagnostic information related to the mobile devices or computers internal to the corporate network." SAMSUNG-1005, ¶¶54, 184, 188, 193, 203-206, 213-216. Additionally, network parameters and different aspects of security features (e.g., values associated with each policy in stored data objects) are stored in device memory 220, 242. SAMSUNG-1005, ¶¶61, 67, 184, 283-293, 239, 192; SAMSUNG-1003, ¶201. Moreover, Wright's diagnostic module collects (and therefore stores, e.g., in memory) diagnostic information and receives diagnostic support information (SAMSUNG-1005, ¶¶60-66, 82-84), which includes software or policy information and which can be used to "repair" or "perform maintenance" on Wright's other modules. SAMSUNG, ¶¶62, 284; SAMSUNG-1003, ¶¶201-205.



SAMSUNG-1005, FIG. 2B (annotated)

Moreover, as explained for claim 4, Wright renders obvious a ***device-assisted services memory partition*** (e.g., memory 220) that stores program code for the one or more device agents, including LSP 814 and policy engine 832. SAMSUNG-1003, ¶206. Additionally, Wright’s client system 201 (which a POSITA would recognize includes Wright’s mobile device) “protects data... in memory 220” that is used for policy management and enforcement. SAMSUNG-1005, ¶¶65, 68-71 (policy setting and enforcement modules leveraging this stored data for their respective policy-related operations), 230, 283; SAMSUNG-1003, ¶¶206-207. A POSITA found obvious for the above-described device state information, which is leveraged by these

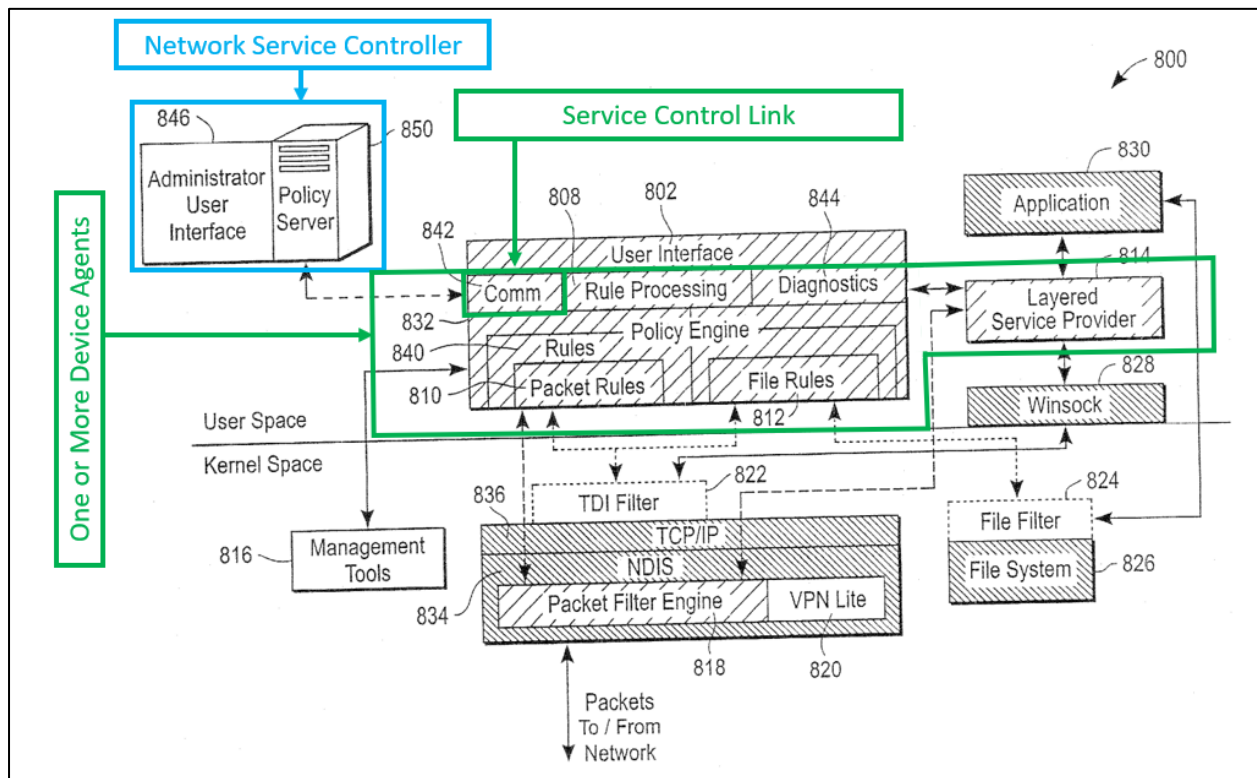
device agents (e.g., for policy management), to be stored in the same memory region where the code for these agents is stored. SAMSUNG-1003, ¶206. A POSITA would so implement Wright's system based on Wright's goal "to protect" such device service state data and because this stored data is utilized by the policy rule engine, for which the code would already be provided securely in the device assisted services memory partition. SAMSUNG-1003, ¶206. Such an implementation would be straightforward and readily implementable. SAMSUNG-1003, ¶206.

Wright renders obvious that *the device state information is only modifiable by the one or more device agents*. SAMSUNG-1003, ¶207. As explained above, the policy engine 832 is the only component in Wright that collects the above-described state information and updates corresponding values in system memory. SAMSUNG-1003, ¶207. Similarly, a POSITA understood Wright's "repair" and "maintenance"-related instructions sent to diagnostic module 844 for updating stored program code are *device service state information*, which is *modifiable by the one or more device agents*. SAMSUNG-1003, ¶207. Wright does not teach or suggest that any components other than policy engine 832 (or its sub-components) as having the privilege to modify the stored device information. SAMSUNG-1014, 69; SAMSUNG-1015, 26, 41, 79; SAMSUNG-1003, ¶207; *cf.* SAMSUNG-1005, ¶66. A POSITA found obvious for the policy engine 832 to be the component with

such *exclusive* modify capability to (1) improve system security by limiting modification capability to certain components, and (2) further Wright’s goal of protecting such device state information (SAMSUNG-1005, ¶¶85-109, 244); SAMSUNG-1003, ¶207. A POSITA would have a reasonable expectation of success in so configuring Wright’s system because a memory partition is already designed to limit access to certain agents, and here, the one or more device agents already have their corresponding code stored in this partition (such that storing data leveraged by such functionality would be straightforward to implement. SAMSUNG-1003, ¶207.

g) Claim 7

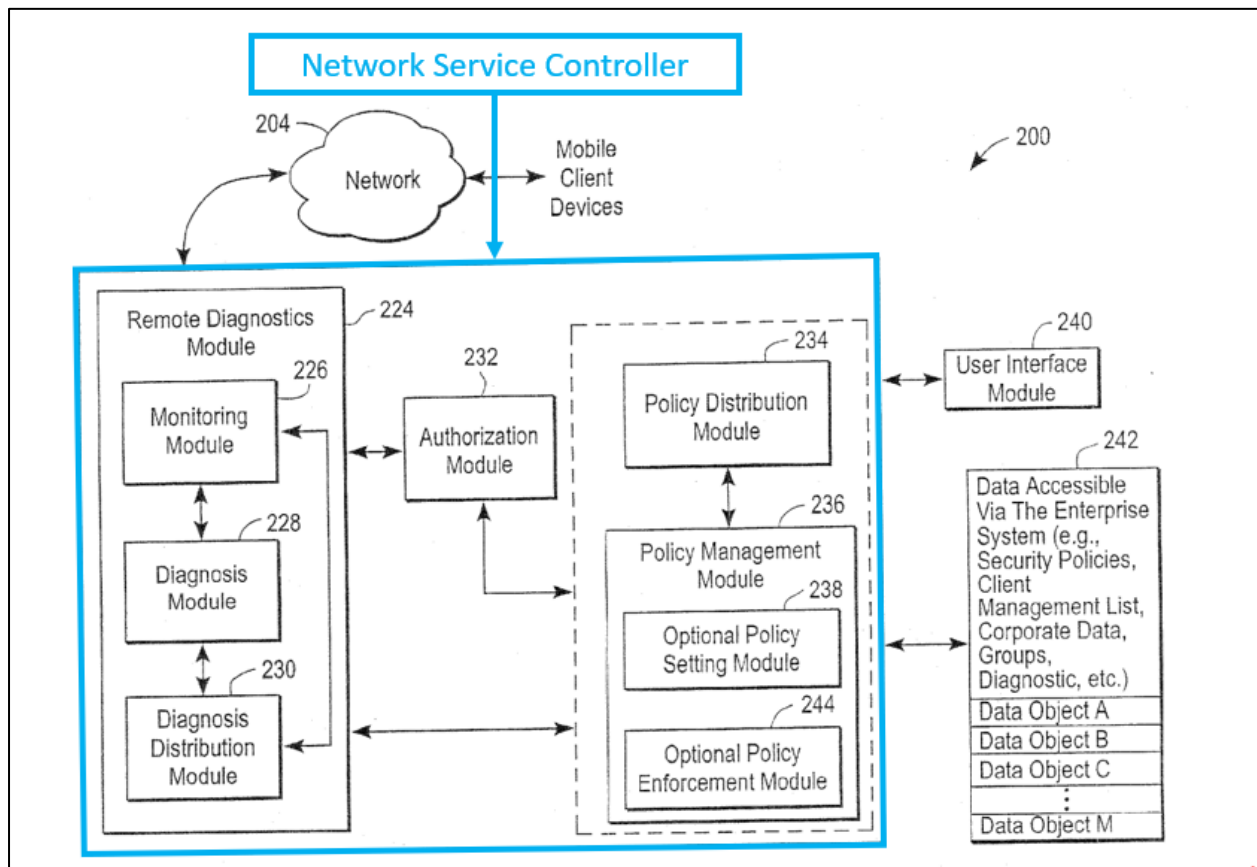
Wright’s “policy server 850 ... is communicatively coupled over a network with the client side system,” and in particular, the policy engine 832, “through a communication port 842.” SAMSUNG-1005, ¶236. As shown below, communication port would be understood or found obvious as being part of policy engine 832 (*one or more device agents*). SAMSUNG-1003, ¶¶208-209, 210; SAMSUNG-1005, ¶¶236, 244; SAMSUNG-1025, 108.



SAMSUNG-1005, FIG. 8 (annotated)

Policy server 850 is a **network service controller** because it is an entity in the network that provides and/or controls network services (e.g., transmission of information over a network, e.g., via device applications) used by the client/mobile device based on security features and policy-related information it provides to the device.

SAMSUNG-1005, Abstract, ¶¶236, 244, 270, 281-287.



SAMSUNG-1005, FIG. 2A (annotated)

Additionally, the communication port 842 is a *service control link* because it is a network link via which the above-described service-related control information (e.g., policy-related security information that controls network communications by the device) are exchanged between the policy server and policy engine 832. SAMSUNG-1003, ¶211.

A POSITA understood that this service control link facilitates *secure communications* given that this link facilitates encrypted communications between policy server 850 and policy engine 832. SAMSUNG-1003, ¶212. As explained above,

policy engine 832 has functionality similar to policy setting module 212, which “determines 338 if the security information” (which includes software or policy information) received from the policy server/server system 200 “is encrypted 338” and if so, decrypts this information “using cryptographic information stored in memory” and previously received “in a secure session.” SAMSUNG-1005, ¶137; *see* SAMSUNG-1001, 6:61-7:31. Because policy engine 832 enables encrypted communications with policy server and such communications happen over communication port 842, a POSITA understood or found obvious that communication port 842 is configured to and enables *secure communications* with the policy server (*network service controller*). SAMSUNG-1003, ¶212. This improves security and proper operation of policy-related functions as managed and enforced on the device and would be straightforward to implement given Wright’s teachings of encrypted network communications. SAMSUNG-1003, ¶¶212-213, 215; SAMSUNG-1005, ¶¶ 66, 169, 247-252, 274-282, 298-300.

The *service control link* (communication port 842) is configured to transmit policy related information (include updates related to authenticated information) to/from policy server via communication port 842. SAMSUNG-1005, ¶¶236, 67 (client mobile device “obtains policy updates from the policy server, updates a local or working policy in accordance with the updates,” and “receives software updates from the server and provides its status, usage and error logs to the server”). In light

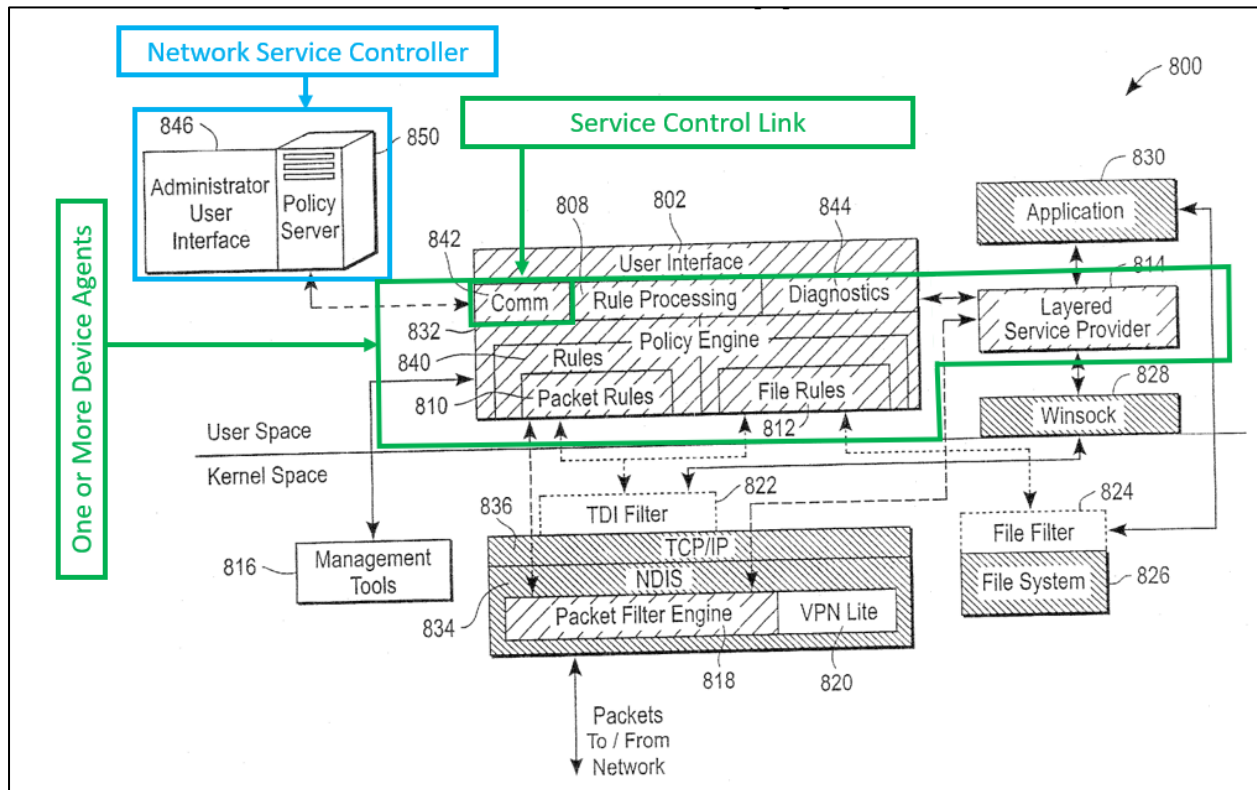
of Wright's disclosure, a POSITA would understand that the *service control link* (communication port 842/communication interface) is configured to receive policies from the policy server 850 and update local or working policies by transmitting information to and from the policy engine 832, including updates related to authentication or encryption. SAMSUNG-1003, ¶¶214-216.

Thus, Wright renders obvious *one or more device agents* (policy engine 832) comprising *a service control link* (communication port 842 alone or in combination with authorization module 245) *configured to securely communicate with a network service controller* (policy server 850). SAMSUNG-1003, ¶¶208-216.

h) Claim 8

Given that (1) Wright's policy setting module "receives, installs and updates the security information including" the above-described policies and other security information (which are *application-specific traffic policy controls*) from a policy server (*network service controller*) over the network (SAMSUNG-1005, ¶¶68, 239, 269; *see supra* [1.7]), (2) Wright policy setting module's functionality is analogously provided by policy engine 832 (SAMSUNG-1005, ¶¶238, 230), and (3) policy engine 832 communicates with the policy server via its communication port 842 (*service control link*) (*see claim 7 supra*), a POSITA understood or found obvious that

Wright's *service control link* is configured to receive at least one of the *application-specific traffic policy controls* from the *network service controller*. SAMSUNG-1003, ¶¶218, 217, 219-223.



SAMSUNG-1005, FIG. 8 (annotated)

i) Claim 9

As explained below, Wright's *service control link* (communication port 842) is *configured to transmit* at least some of the *device service state information* (per claim 7, this includes authorization information, diagnostic information, policies, version numbers, and health checks) to the *network service controller* (policy server 850). SAMSUNG-1003, ¶224.

As explained in [7]-[8], communication port 842 (***service control link***) facilitates communications from policy engine 832 to policy server (***network service controller***). SAMSUNG-1003, ¶225; SAMSUNG-1005, FIGS. 2B, 7-8.

Wright teaches that diagnostic information described in claim 7 (***device service state information***) are transmitted to the policy server's diagnostic module. SAMSUNG-1005, ¶61 (providing examples of diagnostic information), 188-190, 27; SAMSUNG-1003, ¶¶226-227; SAMSUNG-1005, ¶61 (providing examples of diagnostic information), 188-190, 27; SAMSUNG-1001, 17:25-58 (providing similar examples of similar device service state information).

Consistent with the '144 Patent, Wright also discloses transmitting, to the policy server (and in particular, its remote diagnostic module), ***device service state information*** that includes "client configuration information" including device's "current version of the security policy software," "current policy settings," and "attributes in accordance with those settings." SAMSUNG-1005, ¶¶64, 82; SAMSUNG-1001, 7:21-25; 9:64–10:14.

Wright additionally teaches transmitting device status information to the policy server, including sending "status request [that] provides information regarding the client mobile device," such as "authorization information, policies and their version numbers resident on the device, and diagnostic information." SAMSUNG-1005, ¶134; *see id.*, ¶85 (transmitting "encrypted token" from server to device during

trusted connection/session therebetween); SAMSUNG-1003, ¶¶226-232.

j) Claim 10

Policy engine 832 (*one or more device agents*) stores (*retains*) the *device service state information* (e.g., authorization information, diagnostic information, policies, version numbers, and health checks, per claims 6-9 *supra*), which it transmits to the policy server (*network services controller*, per claim 9). SAMSUNG-1003, ¶¶233, 224-239; SAMSUNG-1005, ¶¶81-83, 285, 290-293, 185, 188-193, FIG. 2A-2B.

A POSITA understood or found obvious that the above-described transmitted *device service state information* would be retained in the *one or more device agents* (policy engine 832) *until after an instruction is received that the device service state information can be deleted*. SAMSUNG-1003, ¶237. Per Wright, “client mobile device[] sends,” to the server system..., a status request including device information, e.g., “authorization information, policies and their version numbers resident on the device, and diagnostic information.” SAMSUNG-1005, ¶134. If the server determines updated information is available for the device, (e.g., new policy, version, or software), the server “distributes 327 the updated security information to the mobile device over the network (e.g. Internet) 204” (*id.*) and this security information is then “receive[d], install[ed] and update[d]” by the device’s policy setting

module 212 (*see* SAMSUNG-1005, ¶68).

Because the previously-stored and transmitted information (*transmitted device service state information*) is updated with the newly-received security information, a POSITA understood or found obvious that this update would result in deletion/overwriting of the previously-stored information with the newly-received information. SAMSUNG-1003, ¶237. Such receipt of the newly-received and updated security information is akin to an instruction by the server system/policy server acknowledging that the previously transmitted and retained information can be deleted (e.g., by being overwritten or updated with new information). SAMSUNG-1003, ¶237.

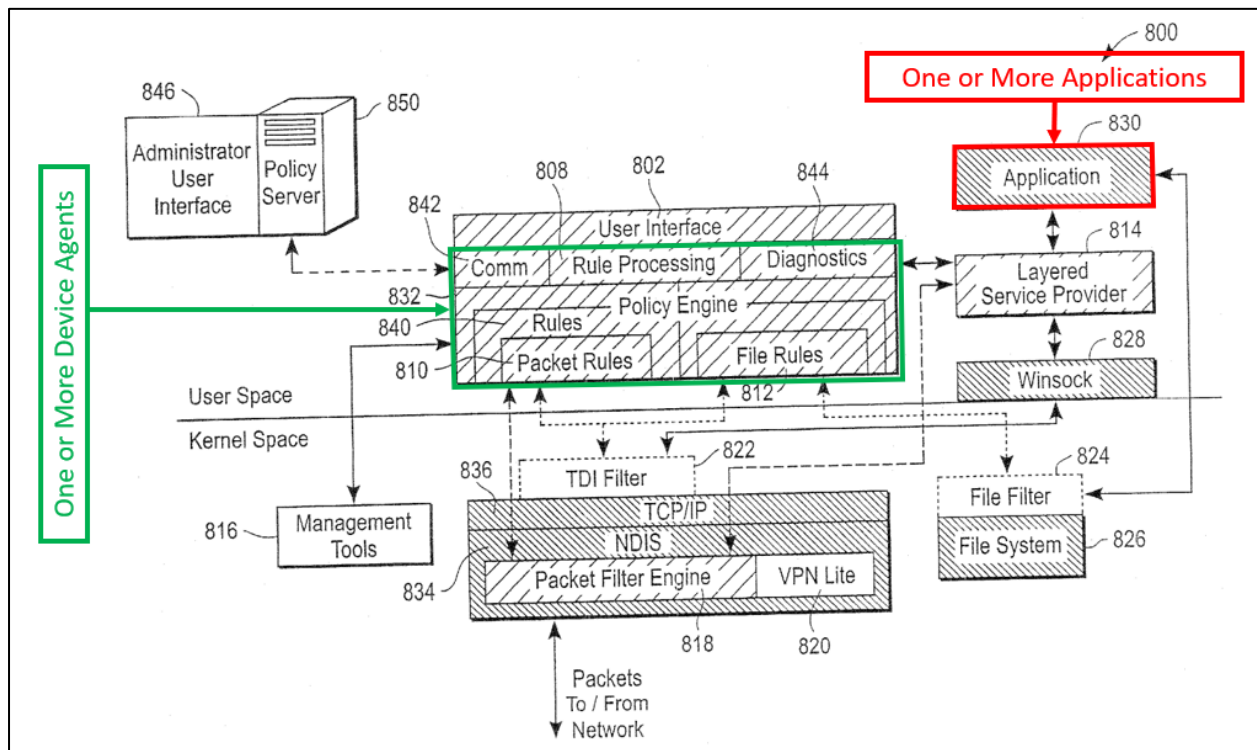
Additionally, Wright teaches diagnostics module (part of policy engine 832; *one or more device agents*) provides “status, configuration, error logs, audit logs, and debug information to the server system” (SAMSUNG-1005, ¶¶285, 270; FIGS. 2A, 8, 11), which then provides “support information” (e.g., code or instructions) for “repair[ing]” the “problem in the client device” or “determin[ing]... preventive maintenance” for the client device. SAMSUNG-1005, ¶¶62, 82. A POSITA understood that “support information,” which includes “instructions or code” for the software update, would delete, overwrite, or alter previous versions or a portion of the installed software. SAMSUNG-1003, ¶283; SAMSUNG-1005, ¶62. Therefore, and as explained above, receiving newly-received instructions or software from the

server system/policy server is akin to *an instruction acknowledging* that the previously transmitted and retained diagnostic *information can be deleted*. SAMSUNG-1003, ¶¶239, 233-239.

k) Claim 11

As described above, Wright's policy engine 832 (*one or more device agents*) configures *application-specific traffic policy controls* for the mobile device. SAMSUNG-1003, ¶240; *see supra* [1.7]-[1.9].

As explained below, a POSITA understood or found obvious that policy engine 832 configures, *based on the above-described application-specific traffic policy controls, application settings for the one or more applications* (e.g., permissions to modify policies, permissions to modify settings, such as modifying visual settings, creating/modifying global objects, allowing access to a specific file). SAMSUNG-1003, ¶241; SAMSUNG-1005, ¶¶55, 68, 72, 85-109.



SAMSUNG-1005, FIG. 8 (annotated)

As explained in [1.7]-[1.8], **application-specific traffic policy controls** include Wright’s application-specific traffic rules and policies, which specify permissions or criteria “relat[ing] to the allowable modification” or overriding that can be made/done to the policy, parameter, or client software. SAMSUNG-1005, ¶¶85, 93, 106-107. An example of a such a policy-specific permission to adjust policy settings includes, e.g., an “[a]bility to modify visual settings.” SAMSUNG-1005, ¶94; SAMSUNG-1003, ¶242.

Because Wright’s policies are application-specific, a POSITA understood or found obvious that a permission to adjust visual settings for a policy would be an

application setting (i.e., a visual setting for an application) that is configurable based on *application-specific traffic policy controls* (Wright's application-specific policies). That's because a POSITA understood that adjusting visual settings, in an application context, would entail, e.g., adjusting appearance of text, images, or other visual attributes displayed in an application. SAMSUNG-1003, ¶243.

Additionally, *application-specific traffic policy controls* also correspond to application-specific rules or policies as they relate to attributes of a file or object, such as size, type, etc., that may be transferred by applications. SAMSUNG-1005, ¶¶158-163, 169; *see supra* [1.6]. Because (1) Wright describes application-specific policies can be based on file attributes of files being transferred and (2) Wright's policies have permissions allowing modification of certain policy settings, a POSITA would have found obvious to implement policies with permissions to modify settings (*application settings*) for the application-specific policies based on attributes of files being transferred by applications (*based on application-specific traffic policy controls*). SAMSUNG-1003, ¶244. In the context of Wright and what was known in the art, application-specific data transfer policies would include configurable permissions to modify attributes (e.g., attributes of file(s) being transferred by applications). SAMSUNG-1003, ¶¶245-246, 58-65; SAMSUNG-1019, 41, 44, 50. Indeed, user-configurable policies enable flexibility in policy implementation

and enforcement for particular scenarios. SAMSUNG-1003, ¶¶246-247. Thus, implementing such file attribute based permissions as part of application-specific policies would have amounted to a known technique to known systems to achieve the predictable results (configurable application settings for application-specific policies) that would be straightforward to implement. SAMSUNG-1003, ¶¶247, 240-246.

I) Claim 13

As described for [1.2]-[1.3], Wright teaches or renders obvious Internet data pack traffic passing from the operating system network stack to the WWAN modem. SAMSUNG-1003, ¶¶248-249.

A POSITA found obvious, in view of Wright's teachings, that Wright's policy engine 832 (*one or more device agents*) would be *configured to encrypt at least some of this Internet data packet traffic*. SAMSUNG-1003, ¶¶248-253. In Wright, the *wireless end-user device* (client system 201) has an authorization module 245 that receives, stores, and transmits encrypted data (*e.g.*, an encrypted file or token). SAMSUNG-1005, ¶¶66, 85, 135, 222. The "authorization module 245 provides authorization information to the authorization module 232 of the server system 200," which a POSITA understood is Internet packet data traffic (since it is data traffic sent as packets over the Internet to the server system 200). SAMSUNG-1005, ¶66;

SAMSUNG-1003, ¶250.

Wright teaches the device sending a “status request” to the server system 200 “checking if new security information is available along with authorization information.” SAMSUNG-1005, ¶135. Wright’s “authorization module 245 may perform the cryptographic authentication protocol,” which includes “employing private keys, public keys or both” as well as “Secure Sockets Layer (SSL) certificates.” SAMSUNG-1005, ¶222, ¶85 (“encrypted token”); SAMSUNG-1046, 1:38-42; SAMSUNG-1047, 3.

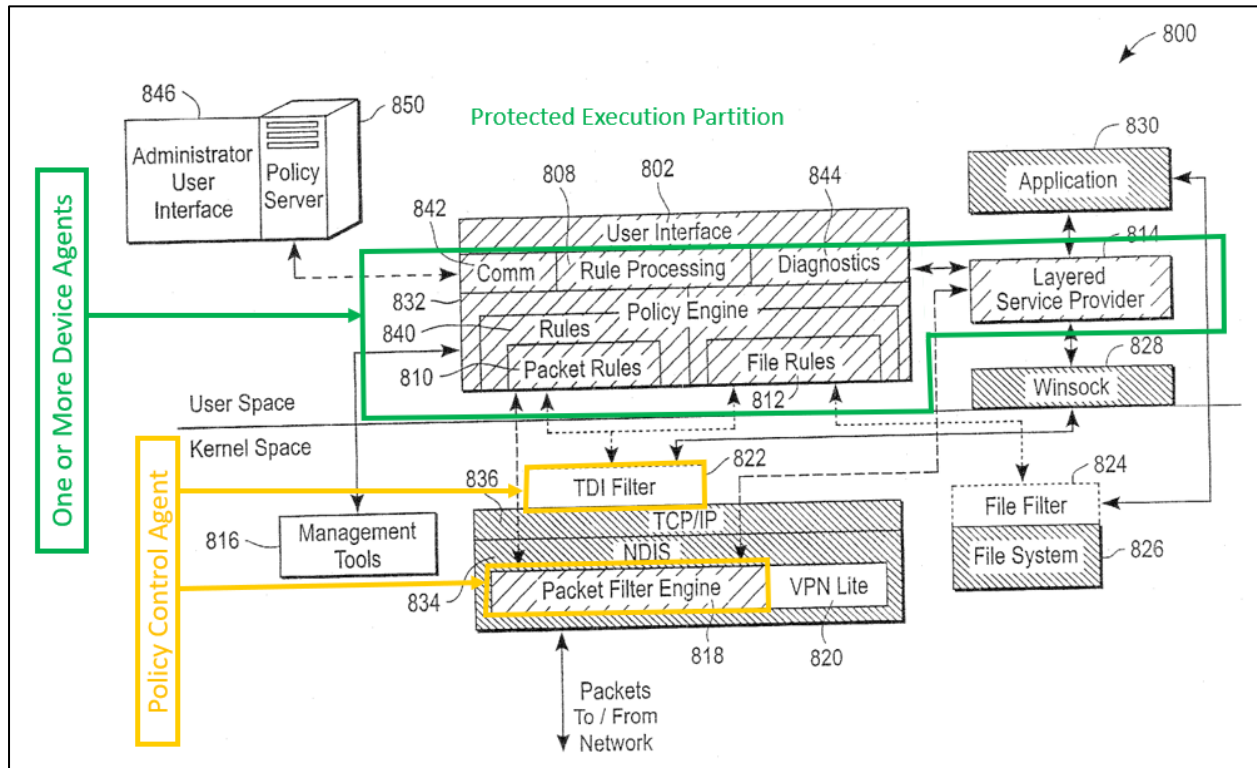
Given these disclosures, a POSITA found obvious that the above-described packet data traffic would be encrypted according to a cryptographic protocol. SAMSUNG-1003, ¶¶251-252. Additionally, it would be obvious to implement the authorization module as part of the diagnostic module of Wright’s policy engine 832, as explained in [5].

Based on the above, Wright’s diagnostic module, which is part of policy engine 832 (*one or more device agents*) would be configured *to encrypt Internet data packet traffic from the operating system packet network stack to the WWAN modem*. SAMSUNG-1003, ¶253.

m) Claim 15

As explained in claim 1, *one or more of the device agents* (policy control

engine 832) has a *privilege to configure the application-specific traffic policy controls* (rules related to application-specific filtering or application-specific network access) of *the policy control agent* (packet filter engine 818, TDI filter 822). See *supra* [1.9]; SAMSUNG-1005, ¶¶268-269, FIG. 8; SAMSUNG-1003, ¶¶254-257.



SAMSUNG-1005, FIG. 8 (annotated)

A POSITA understood that the above-described *privilege to configure the application-specific traffic policy controls* is an *exclusive* privilege. SAMSUNG-1003, ¶255. Per Wright, policy engine 832 is the only entity that (1) receives/downloads the policies from the policy server (SAMSUNG-1005, ¶¶68, 244), and (2) pro-

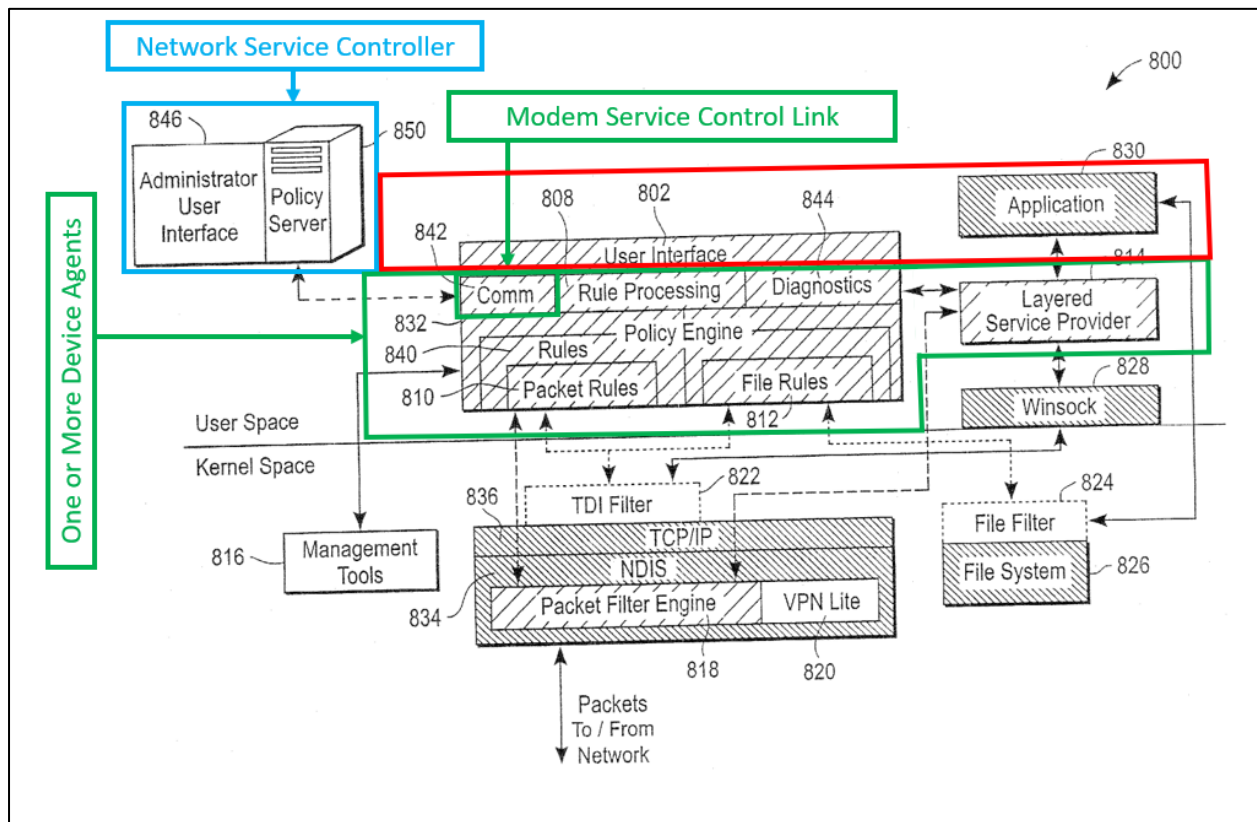
vides the rules underpinning packet filtering by the TDI filter and packet filter engine, for policy enforcement (*id.*, ¶¶239, 245; *see supra* [1.7]-[1.9]). Because no other Wright component has the privilege to receive and configure policies that are enforced by the packet filter engine and the TDI filter, a POSITA understood that this is an exclusive privilege of Wright's policy engine. SAMSUNG-1015, 26-27, 41, 79, 108; SAMSUNG-1014, 69; SAMSUNG-1022, 1283; SAMSUNG-1003, ¶255.

Alternatively, a POSITA found obvious for the policy engine 832 to have the exclusive privilege to receive and configure the application-specific policies. SAMSUNG-1003, ¶¶256-257. A POSITA would have been motivated to limit or make exclusive the policy configuration privilege of the policy engine 832 to ensure that policies received from the policy server are not corrupted or tampered with (SAMSUNG-1005, ¶138), and thus ensuring proper operation of the mobile device. SAMSUNG-1003, ¶256. Additionally, it was well-known in the context of policy management systems, for a single policy setting entity to perform the policy setting and configuration of the policies that are to be enforced, e.g., by a policy enforcement entity. SAMSUNG-1003, ¶257; SAMSUNG-1005, ¶¶235-287, FIG. 8; SAMSUNG-1044, ¶40, FIG. 2. Thus, configuring policy engine 832 to have exclusive privilege to perform the policy configuration would have amounted to implementing

a known technique to a known system to achieve predictable results and functionality, as already contemplated by Wright's system, that would be readily implemented in the system. SAMSUNG-1003, ¶257.

n) Claim 18

Wright's *wireless end-user device compris[es] a user interface*. Wright teaches sending information and instructions, via a user interface module, to the mobile device's "screen." SAMSUNG-1005, ¶¶82-83. Wright also provides examples of "graphical user interface[s]" that are displayed to the user and with which the user interacts, e.g., using an input device (keyboard, mouse, touchpad). *See id.*, ¶¶27-34, 182-193, FIGS. 4-5. In some cases, Wright's "user interfaces" are "displayed for use by a user for establishing criteria for defining an aspect of a security policy as allowed by permissions...." SAMSUNG-1005, ¶¶72, 82 (displaying debug information). A POSITA therefore understood that Wright's mobile device includes a user interface. SAMSUNG-1003, ¶258.; SAMSUNG-1005, ¶¶82-83; FIGS. 4A-5F, ¶¶181-192; SAMSUNG-1043, 607, FIG. 1.



SAMSUNG-1005, FIG. 8 (annotated)

Wright's user interface module (218, 702, 802) corresponds to *a user-interface agent that is configured to display to a user at least one policy setting* (permissions and other user-configurable aspects of policies) *affecting at least one of the application-specific policies* (application-specific policies). SAMSUNG-1005, ¶228; SAMSUNG-1003, ¶259. Wright's user interface module 218 enables "defining an aspect of a policy" (SAMSUNG-1005, ¶186) and "receives input from a user input device ... and causes user interfaces to be displayed for use by a user for establishing criteria for defining an aspect of a security policy as allowed by permissions...." SAMSUNG-1005, ¶72, 236. User interface module 218 is configured to

set “the permissions” that “typically relate to the allowable modification that may be made to a downloaded policy or client software by the client mobile device.” SAMSUNG-1005, ¶¶85, 186. Similarly, in the context of FIG. 8, “policy engine 832 receives user input and sends output via a communication interface with the user interface module 802 to display ... policy settings.” SAMSUNG-1005, ¶244.

Wright’s *user-interface agent* (user interface agent) *is configured to allow modification to at least one policy setting* (permissions) *affecting at least one of the application-specific policies* (application-specific policies). In Wright, “policy setting module 212 may define criteria or if permissions set by the policy management module 236 allow, supplemental policy definitions or customization of policy definitions based upon user input processed by the mobile device user interface module 218.” SAMSUNG-1005, ¶68; *see id.*, ¶¶85, 181, 186. Similarly, “policy setting module 212 defines an aspect of a policy such as location criteria or security features criteria based upon user input processed by user interface module 218.” *Id.* Policy engine 832, which provides functionality analogous to that of policy setting module 212, similarly “receives user input and sends output via a communication interface with the user interface module 802 to ...change policy settings responsive to user input.” SAMSUNG-1005, ¶244.

Wright explains that the user interface module has a communication interface

to multiple system components, including policy setting module 212 (which provides functionality analogous to policy engine in FIGS. 7-8). SAMSUNG-1005, ¶¶72, 244, 228, 186, 188. A POSITA understood the user interface module, together with its communication interface, correspond to one or more *device services assisted agent(s)*. SAMSUNG-1003, ¶260. That's because these components together enable obtaining policy related information and displaying and/or modifying it—thus assisting in the device's performance of such policy management-related services. SAMSUNG-1003, ¶¶261-262.

Additionally, a POSITA would have found obvious to implement these components so that they *are executed within the application execution partition*. SAMSUNG-1003, ¶263. Per Wright, the user interface module is in the user space of the mobile device's environment, which as explained in claim 1 would be partitioned into protected execution partition and an application execution partition. SAMSUNG-1005, FIG. 8. Also, as explained above, the user interface module facilitates displaying of interfaces and the associated interaction of users therewith. A POSITA would thus recognize that the services offered by the user interface module facilitate display and interactions with user interface screens corresponding to policy management functions of the device, and enable such functions for other device applications, e.g., applications 830. SAMSUNG-1003, ¶263. In this regard, FIG. 2B shows the user interface module as a separate component from the group of components (245,

246, 208, 210, 212, and 214) that are responsible for the policy management and diagnostic information collection functionality of the system. *Id.*

Given the general applicability of the user interface module, a POSITA would have sought to implement it as part of the application execution partition and not within the protected execution partition that provides only the policy management and diagnostic-related functions of Wright. SAMSUNG-1003, ¶264. Such an implementation would also be obvious to try since there would be two partitions within user space within which the user interface module could be implemented. SAMSUNG-1003, ¶263. Moreover, user interface agents/modules typically do not need secure environments in which they are implemented and are often implemented in the less secure execution environments within which applications execute. SAMSUNG-1003, ¶263; SAMSUNG-1042, 54. SAMSUNG-1015, 7; SAMSUNG-1025, 42-43, 18. Thus, implementing the user interface module in the execution environment where applications 830 are executed would therefore be an implementation of known techniques to known systems to achieve the predictable result of enabling user interface functionality for all types of device applications (those with higher privileges or not). SAMSUNG-1003, ¶¶258-264.

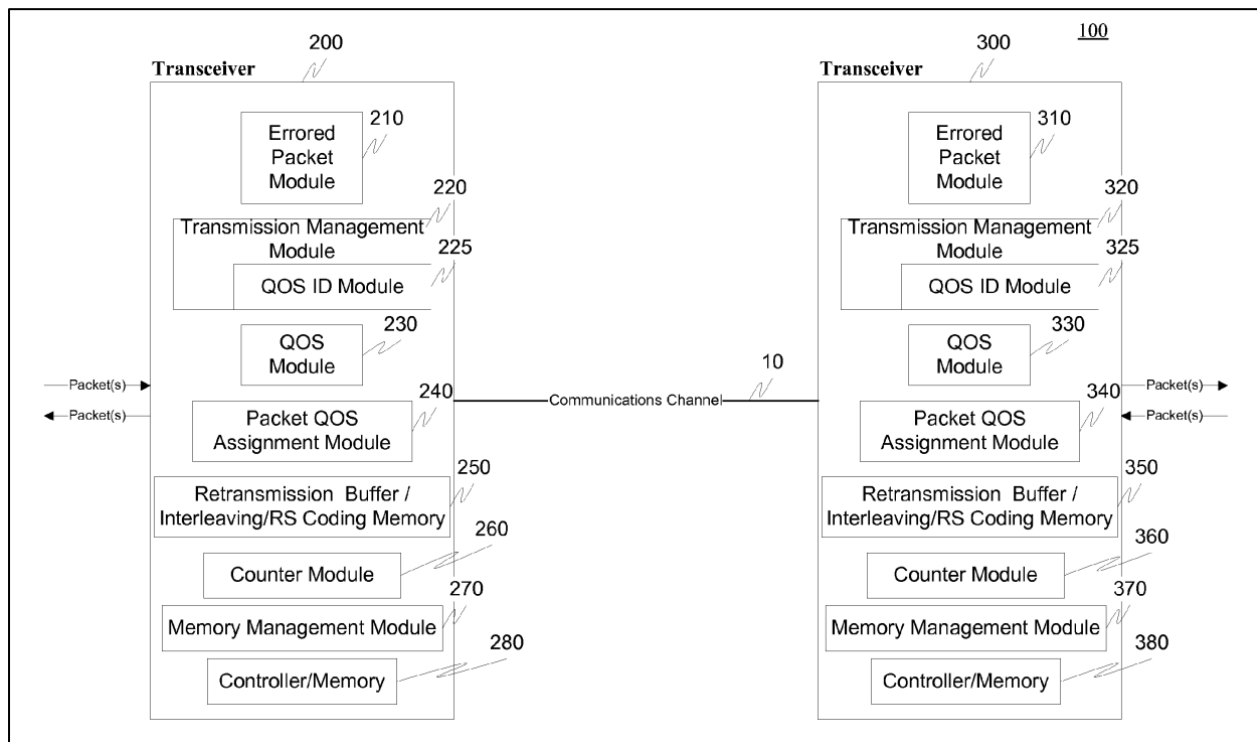
**B. [GROUND 1B] – Wright in View of Tzannes Renders Obvious
Claims 12 and 16-17**

1. Tzannes

Tzannes relates to transmission/retransmission of packet traffic in a communication environment involving computing devices with transceivers/modems.

SAMSUNG-1006, ¶¶2; 185-189, 197; SAMSUNG-1003, ¶¶ 83-88.

Tzannes' communication system 100 is provided below:



SAMSUNG-1006, FIG. 1

Communication channel 10 connects (using wired or wireless link) transceivers/modems 200 and 300. SAMSUNG-1006, ¶¶191, 189. Each transceiver

(200/300) includes “well-known componentry” as well as multiple additional components (210, 220, 225, 230, 240, 250, 260, 270, and 280). SAMSUNG-1006, ¶¶191-192.

Tzannes’ techniques facilitate appropriate transmission/retransmission of packets according to their specified QOS level, which indicates the importance of certain service metrics, e.g., latency and packet error rate (PER), of the packets. SAMSUNG-1006, ¶¶198-204. When transmitting transceiver 200 receives a packet from higher layers, the transceiver’s QOS module 230 can identify packets based on their QOS level, and together with QOS assignment module 240, a QOS identifier is associated with the packet. SAMSUNG-1006, ¶213. If the QOS identifier indicates a low PER packet, transmission management module 220 forwards the packet for storage in retransmission buffer and subsequently transmits the packet. SAMSUNG-1006, ¶214. Alternatively, if the QOS identifier indicates a low latency packet, the packet can be transmitted to the receiving transceiver/modem using the transmission management module 220. *Id.*

“For successful retransmission, the receiving modem should be able to inform the transmitting modem which packet, or packets, need to be retransmitted,” and this is done, e.g., using a packet counter that computes and appends a bit field in the packet that includes a counter (sequence ID; SID) indicating the place of each packet in a stream of packets. SAMSUNG-1006, ¶216. In some instances, packets received

from a higher layer can already have the packet count information. *Id.*, ¶217.

At the receiving transceiver/modem 300, if there is an error in transmission, the missing or error packets can be identified based on the packet count values included in the received packets and corresponding requests can be made to the sending transceiver/modem for packet retransmission. SAMSUNG-1006, ¶¶220, 226-227, 230-248.

2. Combination of Wright and Tzannes

Wright teaches network packet transmissions between a mobile device and one or more components in the network. *See generally* SAMSUNG-1005; §III.A.1 *supra*. In the same field of network communication systems, Tzannes similarly teaches packet communications between computing devices over a network. *See generally* SAMSUNG-1006; §III.B.1 *supra*.

Tzannes additionally enables retransmission of lost/dropped packets based on packet-specific QOS policies and using components/functionality implemented at the sending and receiving transceivers/modems. *See supra* §III.B.1. As explained in §III.B.1, Tzannes teaches packet transmission flows based on QOS levels applied to particular packets and provides a packet counter value in the packet to facilitate packet retransmission. SAMSUNG-1003, ¶90. As to the latter, Tzannes teaches (1) a packet counter module implemented at a transceiver/modem, which calculates a counter value that is appended to the packet and (2) additionally or alternatively,

inserting such computed packet counter values by higher layers. SAMSUNG-1006, ¶¶216-217.

Given that the packet count values could be provided by higher layers before the packet is provided to the modem for transmission, a POSITA recognized or found obvious that Tzannes packet counter functionality can be implemented both at the modem and at higher layers within a computing device. SAMSUNG-1003, ¶90.

A POSITA would have found obvious to combine the above-described teachings of Tzannes' QOS-based packet transmission/retransmission functionality with those of Wright. SAMSUNG-1003, ¶¶90-91. In the resulting Wright-Tzannes combination, Wright's system (as described in §§III.A-B) would implement the functionality and components recited in Tzannes (including at the transceiver/modem and at higher layers), and would facilitate retransmission of lost or dropped packets to a receiver. SAMSUNG-1003, ¶91. As explained below, a POSITA would have been motivated and expected reasonable success in so implementing Tzannes' functionality.

First, Wright recognizes the importance of “alleviat[ing] [network] congestion” (SAMSUNG-1005, ¶255), and in the relevant timeframe, it was well-known that network congestion often resulted in lost or erroneous packet transmissions, which are remedied by retransmission of the lost or erroneous packets. SAMSUNG-

1003, ¶¶89-93; SAMSUNG-1006, ¶¶228, 226-227, 261-263, 283. Combining Tzannes' teachings with those of Wright would therefore provide a more resilient packet transmission system that accounts for network congestion and ensures any lost or undelivered packets are delivered, particularly when QOS rules/service levels call for such delivery. *See* SAMSUNG-1006, ¶218; SAMSUNG-1003, ¶93.

Second, a POSITA found it beneficial to implement Tzannes' traffic management technique that prioritizes certain types of network packets at a mobile device (SAMSUNG-1006, ¶200) to minimize packet loss for certain device applications, e.g., by prioritizing bandwidth for real-time applications while ensuring low latency for applications like gaming and VOIP applications. SAMSUNG-1003, ¶94. Indeed, a POSITA would have sought Tzannes' QOS-based techniques account for service metrics, e.g., latency and PER—to enable prioritizing delivery/transmission of certain types of packets sought to be delivered in Wright's system, e.g., those marked with a QOS level indicating low packet error rate, and facilitating retransmission of the same, e.g., using packet counter values embedded within the packet (as taught by Tzannes). *See* SAMSUNG-1006, ¶5; SAMSUNG-1003, ¶¶94-95; SAMSUNG-1033, 553.

Third, in the relevant timeframe, packet transmission and retransmission subject to Quality of Service (QOS) rules was known to provide a framework by which network traffic was controlled for different scenarios, particularly where network

capacity is limited. SAMSUNG-1003, ¶¶96; SAMSUNG-1020, 128; SAMSUNG-1028, Abstract. Similarly, Tzannes teaches packet transmission/retransmission techniques based on QOS service rules applicable to each packet and Wright contemplates “alleviating” network congestion and provides policies governing a device’s network packet transmissions. *See* §§III.A,1, III.B.1, *supra*. A POSITA would have thus been motivated to combine well-known QOS policies and associated packet transmission/retransmission functionality with Wright to achieve predictable results of enforcing QOS policies, per Tzannes, for network packet transmission (per Wright). SAMSUNG-1003, ¶¶96-97.

Fourth, Wright teaches or renders obvious a WWAN modem for transmission of packet traffic over a network. Given that Tzannes contemplates its functionality being implemented at a modem/transceiver and that its techniques can be implemented for any type of computing device or communication system (SAMSUNG-1006, ¶¶185-197), a POSITA would have had expected reasonable success in implementing components and associated functionality in Tzannes’ modem/transceiver as part of Wright’s WWAN modem. SAMSUNG-1003, ¶98. It would have therefore been straightforward and well-within a POSITA’s skill level to implement what Tzannes already teaches—i.e., components, including the packet counter module, as part of modem (i.e., Wright’s WWAN modem)—to enable, among other functionality, computing and inserting packet count values in a packet’s packet counter field.

SAMSUNG-1003, ¶98.

Moreover, Tzannes also contemplates higher layers of computing devices (higher than modem level) including packet count values. SAMSUNG-1006, ¶¶198, 212; SAMSUNG-1003, ¶99. Wright similarly contemplates higher layer processing of policies, given its disclosure of policies being configured by policy engine 832 and LSP 814 determining and providing of network parameters (e.g., at OSI layers 6 and 7) in some implementations—both of which are in the application/user space and at device’s higher layers. SAMSUNG-1005, ¶¶236-281, FIG. 8. It would thus be well-within a POSITA’s skillset to configure QOS specification and packet count functionality at higher layers, per Tzannes, within components of Wright’s system (e.g., LSP 814) that similarly operate at higher layers and are tasked with similar policy configuration and network analysis/parameter determination. SAMSUNG-1003, ¶99.

3. Claim Analysis

o) Claim 12

As described in §III.B.2, Wright-Tzannes would, in some implementations, include packet counting functionality at LSP 814 (*one or more device agents*) and that such packet counting functionality (as provided by packet counter module 260 in Tzannes and as implemented at LSP 814 in the combination) includes calculating

a packet count value that is appended to the packet being transmitted. SAMSUNG-1003, ¶265; SAMSUNG-1006, ¶¶212, 216.

Packet count calculations corresponds to a *service usage measurement* for the transmitted packets because they calculate each packet's place in the packet stream (i.e., amount of transmitted traffic) being transmitted. SAMSUNG-1003, ¶266. This calculation as implemented at LSP 814 is *separate from* the *traffic information* measured by the TDI filter/packet filter engine (*see supra* [1.6], [1.10]); SAMSUNG-1021, 4; SAMSUNG-1027, 3, 25-26. SAMSUNG-1003, ¶267.

Moreover, a POSITA understood that this packet count calculation is a service usage measurement *for the Internet data packet traffic passing between the operating system packet network stack and the WWAN modem*. SAMSUNG-1003, ¶267. That's because (1) the packet traffic being transmitted in Wright is Internet data packet traffic transmitted from the OS packet network stack to the WWAN (*see supra* [1.2]), and (2) Tzannes, in combination, teaches that transmitted packets received, from higher layers, are delivered externally via the modem/transceiver. SAMSUNG-1003, ¶¶267-268; SAMSUNG-1006, ¶¶192-216.

p) Claim 16

As explained in §III.B.2 and [12] *supra*, Wright-Tzannes leverages Tzannes

packet counting functionality (in packet counter module 260), which enables calculating a packet count value that is appended to the transmitted packet and corresponds to a *service usage measurement*. SAMSUNG-1003, ¶269. Additionally, this packet count value, which is different and *separate from the traffic amount* as described in element [1.6], is a measurement *for the Internet data packet traffic passing between the operating system packet network stack and the WWAN modem*, as explained in claim 12 *supra*. SAMSUNG-1003, ¶269.

Moreover, as explained in §III.B.2, Wright-Tzannes implements components of Tzannes modem/transceiver, including the packet counter module 260, as part of the WWAN modem. SAMSUNG-1003, ¶¶269-270. Additionally, in the relevant timeframe, it was well-known for a mobile device's modem to be implemented using a separate, dedicated processor (i.e., separate from the general processor of the device) with its own dedicated execution environment. SAMSUNG-1003, ¶270. Wright teaches or renders obvious a WWAN modem, which typically includes a specialized processor that is different from a device's general processor. SAMSUNG-1003, ¶270; SAMSUNG-1028, 1-2; SAMSUNG-1031, 748; SAMSUNG-1032, 260. A POSITA would have thus understood or found obvious that Wright's WWAN modem would provide a specialized and separate execution environment (*modem execution partition*), within which the above-described packet counter module would be implemented. SAMSUNG-1003, ¶270.

q) Claim 17

As explained in §III.B.2 and [16] *supra*, Wright-Tzannes' WWAN modem with a ***modem execution partition*** would implement components and functionality of Tzannes' modem/transceiver. SAMSUNG-1003, ¶271. One such component is the transmission management module 220/320 (*see id.*), which is responsible for transmitting packets appended with the calculated packet count values (***service usage measure***, per [16] *supra*) to other computing devices. SAMSUNG-1006, ¶¶212, 214, claims 72, 85, FIG. 1, 192.

Moreover, a POSITA understood that the transmission management module is a ***modem control link*** because it facilitates the communication with external devices (and their respective modems), and controls the packets that are so transmitted by the combination's WWAN modem over communication channel 10, (per Tzannes), which would be akin to the communication channel Wright's communication port 842 and policy server 850. SAMSUNG-1003, ¶272.

Additionally, Tzannes teaches communicating with other device transceivers/modem, just as Wright teaches packet transmission between the device and other network devices, including policy server 850 (***network service controller***, per analysis for claim 7). SAMSUNG-1003, ¶273. In combination, a POSITA understood

or found obvious that the modem's transmission management module facilitates network packet transmissions to the policy server (***network service controller***). SAMSUNG-1003, ¶273. This would have been obvious for reasons described in §III.B.2 *supra* and for the following additional reasons. SAMSUNG-1003, ¶273.

First, a POSITA would recognize that communicating the ***calculated service usage measure*** to the Wright's ***network service controller*** (policy server 850) would enable any subsequent packet retransmission, subject to applicable QOS rules, that may be requested by the policy server 850 (which may not have made it to the server). SAMSUNG-1003, ¶274; SAMSUNG-1006, ¶228.

Second, because Wright's mobile device sends and receives (to/from policy server 850) critical policy information and diagnostic information that controls the operation of Wright's device (*see supra* §III.A-B; SAMSUNG-1005, ¶¶284, 238), leveraging Tzannes' teachings in the combination would facilitate reliable network communications between Wright's mobile device and its policy server (i.e., communications that are not subject to packet loss). SAMSUNG-1003, ¶275.

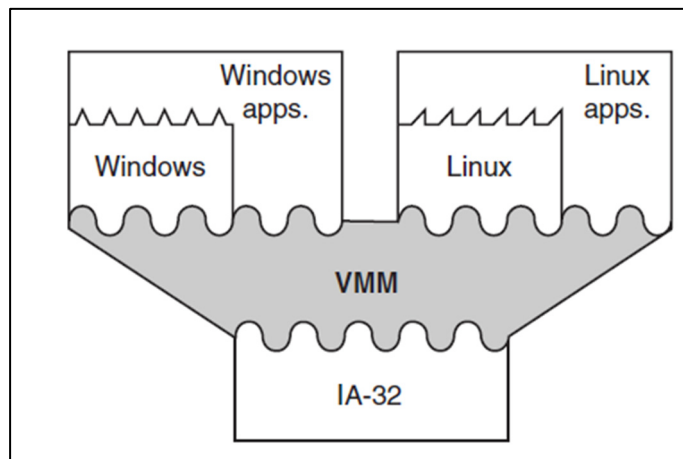
Therefore, Wright-Tzannes renders obvious the ***modem execution partition*** comprising a ***modem control link*** (transmission management module) to communicate ***the calculated service usage measure*** (packet count value) ***to at least one of a network service controller*** (policy server 850). SAMSUNG-1003, ¶276.

**C. [GROUND 1C] – Wright in View of Smith Renders Obvious
Claim 14**

1. Smith

Smith describes various types of virtual machines (VMs), including system VMs, that “provide a complete system environment in which many processes[] can coexist.” SAMSUNG-1007, 23; SAMSUNG-1012, 72. VMs “provide a secure way of partitioning major software systems that run concurrently on the same hardware platform” such that “[s]oftware running” in one VM environment “is isolated from software running on other guest systems.” SAMSUNG-1007, 24; SAMSUNG-1012, 72; SAMSUNG-1003, ¶¶100-109.

“The virtual machine is implemented as a combination of a real machine and virtualizing software,” which may be implemented with “...resources different from the real machine, either in quantity or in type.” SAMSUNG-1007, 15-16. Smith explains that VMs enable “support [of] different operating systems simultaneously” (such as Windows and Linux, as show below). SAMSUNG-1007, 24; SAMSUNG-1012, 72-73.



SAMSUNG-1007/SAMSUNG-1012, FIG. 1.11

2. Combination of Wright and Smith

As explained in §§III.A-B, Wright renders obvious a mobile phone having partitioned environments that receives policies and rules from a remote server. To the extent that Wright does not disclose a secure virtual environment, that would have been obvious based on Smith.

As explained in §III.C.1, Smith teaches implementations of various types of VMs, which “provide a secure way of partitioning major software systems” to isolate software therein. *See* SAMSUNG-1007, 24; SAMSUNG-1012, 72. Per Smith, “[r]unning each application in its own virtual machine increases the robustness of the system,” since erratic behavior on a VM is “less likely to affect the operation” of applications running on a different VM, thereby “provid[ing] isolation and enhanced security.” SAMSUNG-1007, 29; 13; SAMSUNG-1012, 506-507; 56.

A POSITA would have been motivated to improve robustness and security of Wright's device, and to do so, a POSITA would have been motivated to implement Wright's protected execution partition to run in a VM separate from the partitioned environment running the user applications. SAMSUNG-1003, ¶¶102-104, 49-58, 66. The resulting Wright-Smith wireless end-user device includes, a VM Monitor (VMM), which is virtualizing software that provides two VMs, one VM hosting Wright's policy-related engines (policy engine 832, LSP 814), and another VM hosting a second OS running conventional applications, such as Wright's applications 830. SAMSUNG-1003, ¶¶103-104; SAMSUNG-1007, 24; SAMSUNG-1012, 72-73.

A POSITA would have expected success in implementing one or more VMs with Wright, because that would require routine programming knowledge well-within a POSITA's skill. SAMSUNG-1003, ¶105. Indeed, this would have amounted to using a known technique (VMs) to improve operation of a known device (wireless end-user device) to achieve predictable results (running software in a VM). SAMSUNG-1003, ¶105. A POSITA would have found it obvious to implement a portion or all of the protected execution partition in a *secure execution environment* (e.g., a virtual machine (VM)) for at least the following additional reasons.

First, a POSITA would have found it obvious that a VM would provide an

isolated environment that would enable components of the protected execution partition to operate independently without unauthorized access to or from other modules in Wright's user space. A POSITA understood that isolation can be enforced by a VM (hypervisor), and a VM would enable segmentation of computer resources and the ability to manage permissions across VMs. SAMSUNG-1003, ¶106. Using a VM to provide an isolated environment reduces ability for malicious activities in one agents to impact the *secure execution environment* (e.g., the PEP). *Id.* A POSITA understood that isolating environments would facilitate maintaining reliability and confidentiality of data, which is a goal of Wright. *Id.*; SAMSUNG-1005, ¶¶Abstract, 6-17, 285.

Second, a POSITA would have found obvious that a VM would provide a suitable environment for encrypted data. Wright already teaches that one or more device agents, e.g., policy engine 832, implements policies based on a determination of whether a file or data is encrypted. *See, e.g.*, SAMSUNG-1005, ¶¶296, 299-300. A POSITA understood that encrypting data in a VM ensures that sensitive data within a VM is inaccessible to unauthorized users or modules. A POSITA would have found it obvious that a VM could reduce the opportunity for unauthorized access was suitable for controlling access by minimizing user access rights to a minimum requirement for functionality and regular audits of access privileges. SAMSUNG-1003, ¶107.

Third, a POSITA understood that a VM would enable segmenting network traffic. For example, implementing the protected execution partition would enable relative isolation of traffic to/from the PEP to be separated from other parts of the mobile device infrastructure, e.g., Wright's applications 830. Such segmentation minimizes the risk of migration of malicious software or issues that would otherwise compromise security of sensitive, policy management operations of the policy engine 832 and LSP 818 in the PEP. SAMSUNG-1003, ¶¶108-109.

3. Claim 14

As explained in [1.3] *supra*, LSP 814 and policy engine 832 would be implemented as part of *protected execution partition (PEP)*. As explained in §III.C.2, it would be obvious to implement the PEP using one or more virtual machines, thus rendering obvious that the PEP comprises a *secure virtual execution environment*. SAMSUNG-1003, ¶277, 102-109, 50-58; SAMSUNG-1007, 5-6, 34.

IV. TRIAL SHOULD BE INSTITUTED

While discretionary denial is unwarranted here, Petitioner will use the bifurcated briefing process, per Stewart Memorandum, to rebut any contentions by Patent Owner. SAMSUNG-1036; SAMSUNG-1037; SAMSUNG-1038.

V. CONCLUSION AND FEES

The Challenged Claims are unpatentable. Petitioner authorizes charging fees to Deposit Account 06-1050.

VI. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)

Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc. (collectively, “Samsung”) are the real parties-in-interest.

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

The '144 Patent is the subject of civil action *Headwater Research LLC v. Samsung Electronics Co., Ltd. et al*, 2-24-cv-00627, E.D. Tex., filed August 2, 2024 (SAMSUNG-1004). Petitioner is not aware of any other disclaimers, reexamination certificates, or IPR petitions addressing the '144 Patent.

C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Petitioner provides the following designation of counsel.

Lead Counsel	Backup counsel
W. Karl Renner, Reg. No. 41,265 Fish & Richardson P.C. 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 Email: IPR39843-0190IP1@fr.com	Jeremy J. Monaldo, Reg. No. 58,680 Karan Jhurani, Reg. No. 71,777 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 PTABInbound@fr.com

D. Service Information

Please address all correspondence and service to the address listed above.

Petitioner consents to electronic service by email at IPR39843-0190IP1@fr.com
(referencing No. 39843-0190IP1 and cc'ing PTABInbound@fr.com).

Respectfully submitted,

Dated 05/05/2025

/Karan Jhurani/

W. Karl Renner, Reg. No. 41,265
Jeremy J. Monaldo, Reg. No. 58,680
Karan Jhurani, Reg. No. 71,777
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
T: 202-783-5070
F: 877-769-7945

(Control No. IPR2025-00963)

Attorneys for Petitioner

CERTIFICATION UNDER 37 CFR § 42.24

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes* Review totals 13,986 words, which is less than the 14,000 allowed under 37 CFR § 42.24.

Dated 05/05/2025

/Karan Jhurani/

W. Karl Renner, Reg. No. 41,265
Jeremy J. Monaldo, Reg. No. 58,680
Karan Jhurani, Reg. No. 71,777
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
T: 202-783-5070
F: 877-769-7945

Attorneys for Petitioner

CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§ 42.6(e)(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on May 5, 2025, a complete and entire copy of this Petition for *Inter Partes* Review and all supporting exhibits were provided by Federal Express, to the Patent Owner, by serving the correspondence address of record as follows:

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

/Diana Bradley/
Diana Bradley
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
bradley@fr.com