

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Gregory G. Raleigh, et al.
U.S. Patent No.: 9,179,359 Attorney Docket No.: 39843-0181IP1
Issue Date: November 5, 2015
Appl. Serial No.: 14/673,625
Filing Date: March 30, 2015

Title: WIRELESS END-USER DEVICE WITH DIFFERENTIATED
NETWORK ACCESS STATUS FOR DIFFERENT DEVICE
APPLICATIONS

DECLARATION OF DR. KEVIN R. B. BUTLER

I hereby declare that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I further declare that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of the Title 18 of the United States Code.

Dated: Sep 6, 2024

By: 

Kevin R. B. Butler, Ph.D.

TABLE OF CONTENTS

I.	ASSIGNMENT	7
II.	QUALIFICATIONS	7
III.	SUMMARY OF CONCLUSIONS FORMED	12
IV.	LEVEL OF ORDINARY SKILL IN THE ART	13
V.	LEGAL PRINCIPLES	14
	A. Claim Interpretation	14
	B. Priority	15
	C. Anticipation	15
	D. Obviousness	15
VI.	MATERIALS CONSIDERED	17
VII.	TECHNOLOGY OVERVIEW	18
	A. Mobile Devices	18
	B. Resource Management in Mobile Devices	22
VIII.	OVERVIEW OF THE '359 PATENT	23
	A. Specification	23
	B. Claims	26
	C. Prosecution History	30
IX.	INTERPRETATION OF THE '359 PATENT CLAIMS	30
X.	SUMMARY OF RELEVANT PRIOR ART	31
	A. Shell	31
	B. Cole	38
	C. Araujo	42
	D. Freund	44
	E. Montemurro	46
XI.	ANALYSIS	49

A.	GROUND 1A: Shell-Cole Renders Claims 1, 4, 9-11, 13, 14, 18, 19, 31, 23, And 26-28 Obvious.....	49
B.	GROUND 1B: Shell-Cole-Araujo Renders Claims 2, 16-18, 20, 27, and 29 Obvious	105
C.	GROUND 1C: Shell-Cole-Freund Renders Claims 3-5 Obvious.....	119
D.	GROUND 1D: Shell-Cole-Montemurro Renders Claims 6-8, 12, 15, 22, 24, And 25 Obvious.....	127
XII.	CONCLUSION.....	144

EXHIBITS

- SAMSUNG-1001 U.S. Patent No. 9,179,359 to Raleigh (“the ’359 patent”)
- SAMSUNG-1002 Excerpts from the Prosecution History of the ’359 patent (“the Prosecution History”)
- SAMSUNG-1004 Complaint for Patent Infringement in *Headwater Research LLC v. Samsung Electronics Co., Ltd. et al.*, 2-23-cv-00641 (EDTX), Dec. 29, 2023)
- SAMSUNG-1005 U.S. Pat. App. Pub. No. 2004/0224668 (“Shell”)
- SAMSUNG-1006 U.S. Pat. App. Pub. No. 2008/0080458 (“Cole”)
- SAMSUNG-1008 U.S. Pat. App. Pub. No. 2009/0217065 (“Araujo”)
- SAMSUNG-1009 U.S. Pat. App. Pub. No. 2009/0207817 (“Montemurro”)
- SAMSUNG-1010 International Publication No. WO 03/100581 (“Dive-Reclus”)
- SAMSUNG-1011 David Flanagan, O’Reilly & Associates, Inc., “Java in a Nutshell,” 1996, ISBN: I-56592-183-6
- SAMSUNG-1012 U.S. Pat. No. 5,987,611 (“Freund”)
- SAMSUNG-1013 U.S. Pat. App. Pub. No. 2009/0210537 (“Irwin”)
- SAMSUNG-1014 Korhonen, “Host Identity Protocol (HIP) Implementation in the Symbian Environment,” Master of Science Thesis, Tampere University of Technology (12/10/2007) (“Korhonen”)
- SAMSUNG-1015 U.S. Pat. No. 8,886,261 (“Aerrabotu”)
- SAMSUNG-1016 Welsh, “Incorporating Memory Management into UserLevel Network Interfaces.” IEEE Micro, 18(2), 1998, 10 pages (“Welsh”)

- SAMSUNG-1017 U.S. Pat. App. Pub. No. 2009/0093247 to Srinivasan (“Srinivasan”)
- SAMSUNG-1018 U.S. Pat. App. Pub. No. 2006/0039354 (“Rao”)
- SAMSUNG-1019 Carter, Casey et al., “Contact networking: a localized mobility system,” Proceedings of the 1st International Conference on Mobile systems, Applications and Services, 2003 (“Carter”)
- SAMSUNG-1020 U.S. Pat. No. 8,413,172 (“Sng”)
- SAMSUNG-1021 S. Lee & N. Golmie, “Power-Efficient Interface Selection Scheme using Paging of WWAN for WLAN in Heterogeneous Wireless Networks,” 2006 IEEE International Conference on Communications (Istanbul, 2006), 1742–1747 (“Lee”)
- SAMSUNG-1022 M. Stemm et al., “A network measurement architecture for adaptive applications,” Proceedings of the 2000 IEEE INFOCOM Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), Tel Aviv, Israel, 2000, 285-294 (“Stemm”)
- SAMSUNG-1023 K. Ravindran & V. Bansal, “Delay compensation protocols for synchronization of multimedia data streams,” IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No. 4, 574-589, Aug. 1993 (“Ravindran”)
- SAMSUNG-1024 D. C. Verma, “Simplifying network administration using policy-based management.” IEEE Network, Vol. 16, No. 2, 20-26, Mar. 2002 (“Verma”)
- SAMSUNG-1025 G. Nychis & D. R. Licata, “The impact of background network traffic on foreground network traffic.” Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), 2001 (“Nychis”)

- SAMSUNG-1026 I. Pronchev. "Packet Capturing Using the Linux Netfilter Framework." Technical Report, Technische Universität München, 1-31, July 2006 ("Pronchev")
- SAMSUNG-1027 A.-J. Moerdijk and L. Klostermann, "Opening the networks with Parlay/OSA: standards and aspects behind the APIs." IEEE Network, vol. 17, no. 3, pp. 58-64, May-June 2003 ("Moerdijk")
- SAMSUNG-1028 J. B. D. Joshi, E. Bertino, U. Latif and A. Ghafoor, "A generalized temporal role-based access control model." IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 1, pp. 4-23, Jan. 2005 ("Joshi")
- SAMSUNG-1029 Lymberopoulos et al., "An Adaptive Policy-Based Framework for Network Services Management," Journal of Network and Systems Management, Vol. 11, No. 3, September 2003 ("Lymberopoulos")
- SAMSUNG-1030 Richard Stevens et al., "UNIX Network Programming Volume 1, Third Edition: The Sockets Networking API," 2004, ISBN: 0-13-141155-1

I, Kevin R. B. Butler of Gainesville, Florida, declare that:

I. ASSIGNMENT

1. I have been retained on behalf of Samsung Electronics Co., Ltd. (“Samsung”) to offer technical opinions related to U.S. Patent No 9,179,359 (“the ’359 patent”) (SAMSUNG-1001). I understand that Samsung is requesting that the Patent Trial and Appeal Board (“PTAB” or “Board”) to institute an *inter partes* review (“IPR”) proceeding of the ’359 patent.

2. I have been asked to provide my independent analysis of the ’359 patent based on the prior art publications cited in this declaration.

3. I am not and never have been, an employee of Samsung. I received no compensation for this declaration beyond my normal hourly compensation based on my time actually spent analyzing the ’359 patent, the prior art publications cited below, and issues related thereto, and I will not receive any added compensation based on the outcome of any IPR or other proceeding involving the ’359 patent.

II. QUALIFICATIONS

4. I am over the age of 18 and am competent to write this declaration. I have personal knowledge, or have developed knowledge of these technologies based upon education, training, or experience, of the matters set forth herein.

5. I have summarized in this section my educational background, career history, publications, and other relevant qualifications.

6. I received a B.Sc. degree in Electrical Engineering from Queen's University in 1999, an M.S. degree in Electrical Engineering from Columbia University in 2003, and a Ph.D. in Computer Science and Engineering from the Pennsylvania State University in 2010.

7. I am an Associate Professor with tenure in Computer and Information Science and Engineering and Associate Director of the Florida Institute for Cybersecurity Research at the University of Florida. I founded and directed the Oregon Systems Infrastructure and Information Security (OSIRIS) Lab at the University of Oregon, where I was an assistant professor of computer and information science for four years before moving to Florida under the Rising to National Preeminence initiative. I have authored over 115 peer-reviewed publications including book chapters, journal articles, conference proceedings papers, and workshop papers. My research is in the area of computer systems security, including embedded systems and firmware, as well as network security, and with a particular focus on the security of peripheral devices that use the USB protocol and the hosts that these devices connect to. I received a National Science Foundation CAREER award in 2013, an International Educator of the Year award within the Herbert Wertheim College of Engineering at the University of Florida in 2017, a University of Florida Term Professorship for 2018-2021, and an Arnold and Lisa Goldberg Rising Star Endowed Professorship in Computer Science for

2018-2024. I also received an Outstanding Community Service Award from the IEEE Technical Committee on Security and Privacy in 2017. I am a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE), and a Senior Member of the Association for Computing Machinery (ACM). I have been involved with international standards-setting bodies; in particular, I was co-chair and leader of the Security Workstream for the International Telecommunications Union (ITU)'s Technology, Innovation, and Competition Working group within the Focus Group on Digital Financial Service from 2015-2016, and from 2017-present, I serve as Co-Chair of the Security, Infrastructure, and Trust Working Group within the ITU's Financial Inclusion Global Initiative. I currently serve on four journal editorial boards, have served on numerous funding panels for the National Science Foundation and reviewed grant proposals on behalf of science and engineering boards in Austria, Canada, and Israel. I have served on over 115 technical program committees for selecting papers to appear at academic conferences on computer security, and I have served as program committee chair and general conference chair for multiple conferences. I have also given over 100 presentations and invited talks, including invited lectures at the University of Oxford, ETH Zurich, the Chinese Academy of Sciences, Dartmouth College, Columbia University, and UCLA. My research has been featured in news outlets such as the Wall Street Journal, BBC News, and WIRED.

8. I have experience designing and implementing communication protocols, including peripheral device communications. As a research assistant at Columbia University in 2001-2002, I incorporated authentication protocols into a Java-based software stack for the Session Initiation Protocol used for Internet telephony. At Flarion Technologies in 2003, I implemented packet diversion and IP header modification program in the FreeBSD operating system for mobile IP packet testing, to be deployed on WiFi devices. Since then I have led the development of mechanisms to constrain the functionality of USB devices by incorporating user expectations of device capabilities into the services that they can advertise. I also lead the development of protocols for allowing a rule-based approach to constraining malicious functionality in USB, Bluetooth, and near-field communication (NFC) devices.

9. I have experience with the development and analysis of software and firmware. In 1998 at the Royal Bank of Canada, I developed client-server applications, including design of graphical user interfaces on the client side, to port business processes from mainframe environments. Further, in 2006 I compiled and modified OpenEmbedded firmware to support the SlugOS environment to allow experimentation with storage devices, in order to develop a disk prototype for preventing a type of malware called a rootkit from persisting on a system after reboot. Since then, I have led efforts to analyze firmware on embedded devices,

particularly firmware from USB flash drives that uses the Intel 8051 microcontroller, and demonstrated how such binary firmware could be lifted to an intermediate representation to enable control flow graph recovery and symbolic analysis. I also led a project that demonstrated the extraction of data from smartphone firmware images and demonstrated that legacy command sets (i.e., “AT commands” developed for modems in the 1980s) were being used to provide undocumented features on these devices that could be accessed over a USB connection, as well as methods for enabling the transmissions of these commands.

10. I have experience with peripherals and portable devices. In course projects as an undergraduate in 1997 and 1998, I developed code for communication with a Motorola HC11 microcontroller to interface with a host computer. Since then, I have developed storage devices as discussed above that provide security services. Further, in 2009, I developed a USB device capable of attesting the integrity of the host computer it was attached to while at Symantec Research. This work was later patented (U.S. Patent 8,856,918, Host validation mechanism for preserving integrity of portable storage data. Petros Efstathopoulos, Bruce Montague, Dharmesh Shah, Kevin Butler. October 7, 2014.). Since then, I have overseen the development of USB devices that allow for tracking the provenance of data at the block level and the development of mobile device software that automates the collection of USB enumeration data.

11. I have experience with encryption and cryptographic protocols. At Telcordia Technologies in 2000 and 2001, I provisioned and configured cable modems within a hybrid fibre-coax lab that involved secure telephony and the use of encryption over DOCSIS modems. I further led the design of developing cryptographic primitives to support the secure delivery of routing information within the Border Gateway Protocol and have been involved with numerous projects incorporating encryption, such as a mechanisms using additive homomorphic encryption to support accounting audits for preserving the confidentiality of wiretap records and cryptographic schemes for assuring the secure time release of data.

12. My curriculum vitae, attached as Exhibit A, includes a list of publications on which I am a named author. It contains further details regarding my experience, education, publications, and other qualifications provide to an expert opinion in connection with this proceeding.

III. SUMMARY OF CONCLUSIONS FORMED

13. This Declaration explains the conclusions that I have formed based on my analysis. To summarize those conclusions:

- Claims 1, 4, 9-11, 13, 14, 18, 19, 21, 23, and 26-28 are made obvious by Shell and Cole

- Claims 2, 16-18, 20, 27, and 29 are made obvious by Shell, Cole, and Araujo
- Claims 3-5 are made obvious by Shell, Cole, and Freund
- Claims 6-8, 12, 15, 22, 24, and 25 are made obvious by Shell, Cole, and Montemurro

IV. LEVEL OF ORDINARY SKILL IN THE ART

14. Based on my understanding, a person of ordinary skill in the art (“POSITA”) is a hypothetical person who is presumed to have the skill and experience of an ordinary worker in the field at the time of the alleged invention. The ’359 patent was filed March 30, 2015, and claims priority to several provisional applications, the earliest of which was filed January 28, 2009 (“Critical Date”). Because I do not know at what date the invention as claimed was made, if ever, I have used the Critical Date of the ’359 patent as the point in time for claim interpretation purposes. My opinion does not change if the invention date is earlier.

15. Based on my knowledge and experience in the field and my review of the ’359 patent and file history, I believe that a person of ordinary skill in the art in this matter would have had (1) at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or a related field, and (2) at least two years of industry experience in services and application implementation in communication networks. Additional graduate education could substitute for

professional experience, and vice versa. Based on my experiences, I have a good understanding of the capabilities of a POSITA. I have taught, mentored, advised, and collaborated closely with many such individuals over the course of my career.

V. LEGAL PRINCIPLES

16. I am not a lawyer and I will not provide any legal opinions in this IPR. Although not a lawyer, I have been advised that certain legal standards are to be applied by technical experts in forming opinions regarding the meaning and validity of patent claims.

A. Claim Interpretation

17. I understand that claim terms are generally given their plain and ordinary meaning based on the patent's specification and file history as understood by a person of ordinary skill in the art at the time of the purported invention. In that regard, I understand that the best indicator of claim meaning is its usage in the context of the patent specification as understood by a POSITA. I further understand that the words of the claims should be given their plain meaning unless that meaning is inconsistent with the patent specification or the patent's history of examination before the Patent Office. I also understand that the words of the claims should be interpreted as they would have been interpreted by a POSITA at the time of the invention was made (not today).

B. Priority

18. I understand that a continuation application is a later-filed application that has the same disclosure (specification and figures) as an earlier filed application to which the later-filed application claims priority. A continuation is generally entitled to the same priority date as the later-filed application to which it claims priority.

C. Anticipation

19. I understand that a patent claim is invalid as anticipated if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. I also understand that, to anticipate, the reference must teach all of the limitations arranged or combined in the same way as described in the claim. I do not rely on anticipation in this declaration.

20. With respect to inherency, I understand that the fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. Instead, the inherent characteristic must necessarily flow from the teaching of the prior art.

D. Obviousness

21. I understand that a patent claim is invalid if the claimed invention would have been obvious to a person of ordinary skill in the field at the time of the purported invention, which is often considered the time the application was filed.

Even if all of the claim limitations are not found in a single prior art reference that anticipates the claim, the claim can still be invalid.

22. To obtain a patent, a claimed invention must have, as of the priority date, been nonobvious in view of the prior art in the field. I understand that an invention is obvious when the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art.

23. I understand that, to prove that prior art or a combination of prior art makes a patent obvious it is necessary to: (1) identify the particular references that, singly or in combination, make the patent obvious; (2) specifically identify which elements of the patent claim appear in each of the asserted references; and (3) explain a motivation, teaching, need, market pressure or other legitimate reason that would have inspired a person of ordinary skill in the art to combine prior art references to solve a problem.

24. I also understand that certain objective indicia can be important evidence regarding whether a patent is obvious or nonobvious. Such indicia include:

- Commercial success of products covered by the patent claims;
- A long-felt need for the invention;

- Failed attempts by others to make the invention;
- Copying of the invention by others in the field;
- Unexpected results achieved by the invention as compared to the closest prior art;
- Praise of the invention by the infringer or others in the field;
- The taking of licenses under the patent by others;
- Expressions of surprise by experts and those skilled in the art at the making of the invention; and
- The patentee proceeded contrary to the accepted wisdom of the prior art.

25. To the extent these factors have been brought to my attention, if at all, I have taken them into consideration in providing my opinions and conclusions.

VI. MATERIALS CONSIDERED

26. My analysis and conclusions set forth in this declaration are based on my educational background and experiences in the field (*see* Section II). Based on my knowledge and experience, I believe that I am considered to be an expert in the field. Also, based on my knowledge and experience, I understand and know of the capabilities of persons of ordinary skill in the field during the early 2000s–2010s, and I taught, participated in organizations, and worked closely with many such persons in the field during that time frame.

27. As part of my independent analysis for this declaration, I have considered the following: the background knowledge/technologies that were commonly known to persons of ordinary skill in this art during the time before the earliest claimed priority date for the '359 patent; my own knowledge and experiences gained from my work experience in the field of the '359 patent and related disciplines; and my experience in working with others involved in this field and related disciplines. I have analyzed the publications and materials listed on pages 3 and 4 of this Declaration. My analysis and conclusions set forth in this declaration are based on the perspective of a POSITA.

VII. TECHNOLOGY OVERVIEW

28. In this section I provide a broad overview of the relevant technology and state of the technical community at the priority date of the '359 patent. This description is not intended to be exhaustive, but to provide sufficient background to support and contextualize my analysis below.

A. Mobile Devices

29. In 2009, use of smartphones experienced a significant surge, marking a pivotal moment in the evolution of mobile technology. The year saw a growing number of consumers transitioning from traditional feature phones (e.g., flip phones) to more capable and versatile smartphones. Key factors driving this uptake included advancements in mobile operating systems such as Apple's iOS and

Google's Android, offering a user-friendly interface and access to a burgeoning app ecosystem. Additionally, the earlier introduction of 3G networks enabled faster mobile data speeds, prompting users to embrace smartphones for tasks beyond voice and messaging. As a result, smartphone sales soared, laying the groundwork for the transformative impact they would have on communication, productivity, and entertainment in the years to come.

30. However, smartphones in 2009 were resource-limited compared to the current devices. For example, smartphones of 2009 were equipped with single-core processors, usually running at speeds below 1 GHz. These processors lacked the computing power of today's multi-core CPUs found in modern smartphones. As a result, tasks that required significant processing, such as running complex applications or playing graphically intensive games, were often slow and less efficient. Similarly, (non-transitory) memory capacities in smartphones of 2009 were generally limited, typically ranging from 128 MB to 512 MB. Long term storage was also limited, consisting of between 4 and 16GB of space. Devices were

connected to wireless networks using WWAN (cellular) and WLAN (Wi-Fi) modems.¹

31. The networks that were being used in 2009 were also limited in speed and throughput. In 2009 wide-area (WWAN) cellular 3G networks, as implemented by the Global System for Mobile Communications (GSM)² and Code Division Multiple Access (CDMA) protocols were becoming widespread, but 4G networks were still in the early stages of deployment. This meant slower internet speeds and limited coverage for mobile data, affecting the overall online experience. Wi-Fi technology as implemented in the 802.11 suite of protocols complemented cellular networks. Smartphones, laptops, and other devices could connect to Wi-Fi hotspots in homes, offices, cafes, and public spaces to access the internet at much higher speeds than what 3G networks could offer. Wi-Fi also

¹ The term “modem” refers to a modulator/demodulator, which is a device to process electrical signals received and sent over a physical media (e.g., copper wire) or over the air. In mobile devices, modems are software-supported radios that process specific network access protocols.

² Note that GSM predates 3G networks (actually back to 1991), but, for the purposes of this report, I will focus only on those variants of GSM which were classified as 3G.

allowed users to offload data usage from their cellular plans, which was particularly helpful due to the relatively limited data caps on mobile plans during that time. Regardless of the network protocol it was running on, the IP protocol was used to connect each device with the Internet.

32. At the same time, battery technology had not yet seen significant advancements, leading to smartphones with relatively short battery life. The computational requirements and increased user engagement (using the phone for longer periods) led to increased demand for energy. Here, the combination of power-hungry processors, smaller batteries, and less optimized software meant that users often had to charge their phones frequently.

33. While 2009 smartphone displays were already evolving towards touchscreens, they were smaller, less vibrant, and had lower resolutions compared to modern displays. Common screen resolutions were around 320x480 pixels, making it challenging to display content with high visual fidelity. Devices operating systems provided windowed environments that would generally have a single application being displayed (and prioritized by the operating system) and interacting with the users. These applications were referred to as having focus (of the user interface). Users could switch between applications as needed but could only interact with the one application with focus at a time.

B. Resource Management in Mobile Devices

34. A key challenge in 2009 was finding ways to address the rapidly increasing needs of mobile users using the resource-limited devices and networks. Solutions to these problems were developed by adapting known techniques in computer systems and networks (e.g., priority queuing), as well as developing new, faster devices and networks.

35. Policy for computational systems has been a part of the technical landscape since the 1970s, with a primary focus on differentiating resource usage in various environments. For example, a policy might specify that data from interactive applications, such as multi-player games, are assigned the highest priority to ensure the best user experience. These policies were implemented to manage and allocate computing resources efficiently in diverse computing environments. For instance, differentiated services emerged as a crucial policy in this context, especially as mobile devices such as phones began to traverse heterogeneous environments, transitioning between Wi-Fi and cellular networks seamlessly. By employing policies like differentiated services, computing systems could optimize resource allocation, ensuring a smooth and reliable user experience while adapting to changing network conditions and device capabilities in these dynamic environments. Such techniques were particularly important to address the challenges faced by early mobile devices.

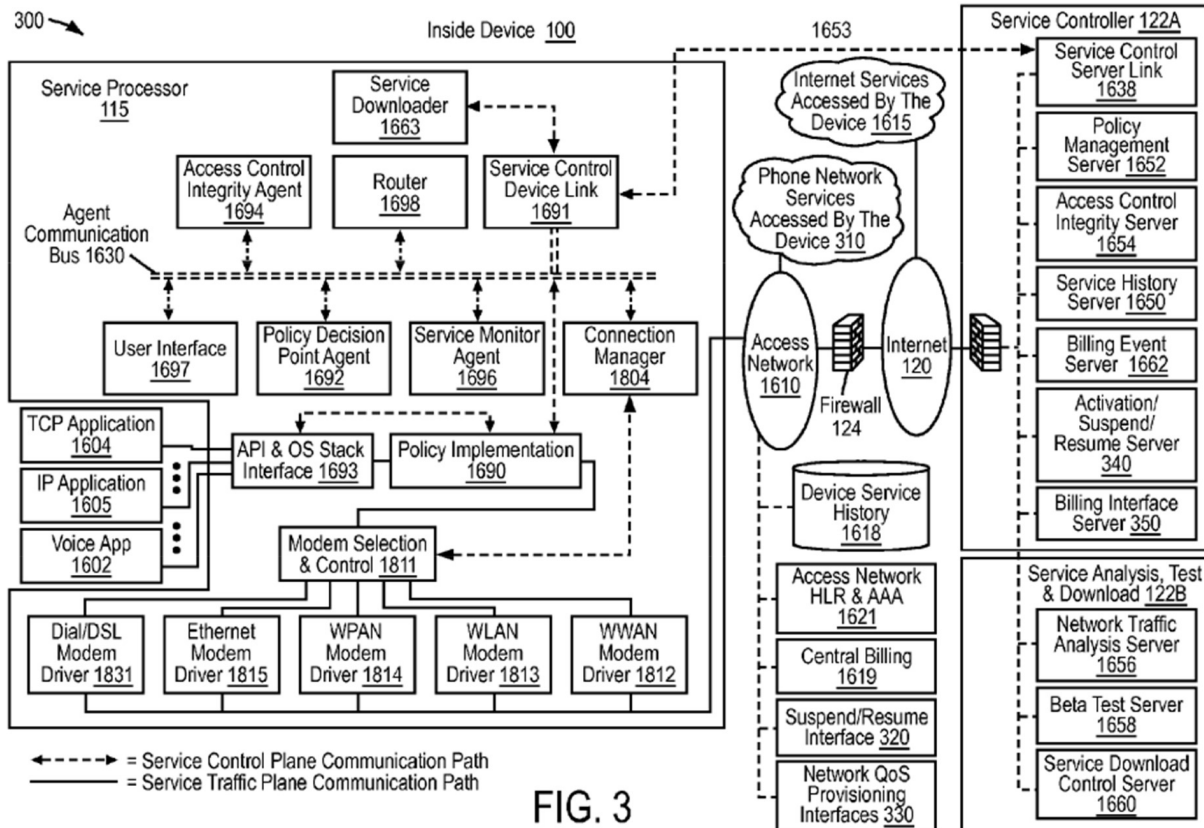
36. A Policy Decision Point (PDP) and Policy Enforcement Point (PEP) are essential components in policy-based systems. The PDP functions as the determining logic of the system, responsible for evaluating policy rules and making decisions based on the input received from sources. Conversely, the PEP serves to enforce policy. PEPs are located at the point where policies are applied. The PEP communicates with the PDP to obtain the relevant policy decisions and then enacts them by controlling access to resources. In the example given above, the PDP would determine whether data are from an interactive game, and if so, the PEP would assign the highest priority to the data, e.g., by adding the data to a high-priority queue. Again, this would ensure the user does not experience performance problems (e.g., lag).

VIII. OVERVIEW OF THE '359 PATENT

A. Specification

37. The '359 patent is directed to “wireless end-user device with differentiated network access status for different device applications.” SAMSUNG-1001, Title. The '359 patent explains that “a wireless end-user device has a wireless wide-area network (WWAN) modem and supports multiple wireless network types,” which can include a WLAN modem, as illustrated below. SAMSUNG-1001, Abstract, FIG. 3. The '359 patent teaches that “in wireless networks, managing the wireless access connection capacity and network access

connection resources is important to maintain network performance as network resources/capacity demand increases.” SAMSUNG-1001, 3:44-4:5.



SAMSUNG-1001, FIG. 3

38. To manage capacity, the device can “apply a service usage control policy for the network service usage activities to protect network capacity (e.g., reduce network capacity demand).” SAMSUNG-1001, 11:16-12:2. The device can “assign[] a priority to a network service usage activity and control[] traffic associated with the network services usage activity based on the assigned priority” and can provide a “differentiated and dynamic background services classification

... as a function of network availability state and/or network busy state.” *Id.*, 12:3-10. The service usage control policy can assist “in network access control of network service usage activities (e.g., deferring some or all of the network capacity demand from these source activities).” *Id.*, 12:11-27. The device can associate “the network service usage activity with a network service usage control policy” based on factors such as “a network busy state, a time of day, ..., an application behavior, ..., a power state of device, a device usage state, a time based criteria, availability of alternative networks, and a set of policy rules.” *Id.*, 85:9- 20. When a policy applies to an access request, the device can take various actions including acting to “throttle, delay, priority queue, time window, suspend, quarantine, kill, remove, and other well known traffic control techniques.” *Id.*, 13:10- 30.

39. To enforce the controls, the device includes “a policy implementation driver/agent 1690 (e.g., policy enforcement point driver/agent)” and a Policy Decision Point Agent 1692. SAMSUNG-1001, FIG. 3, 57:34-58:17. A POSITA would have recognized policy decision points (PDPs) and policy enforcement points (PEPs) as terms of the arts describing components that determine whether a policy applies (PDP), and when the policy applies, enforcing the policy decision (PEP). SAMSUNG-1010, [0020].

B. Claims

40. The table below includes identifiers for the '359 patent claims and limitations, which are referenced throughout.

Claim	Identifier	Claim Language
1	[1.pre]	A wireless end-user device, comprising:
	[1.1]	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;
	[1.2]	a wireless local area network (WLAN) modem to receive and transmit Internet data between the device and at least one WLAN when configured for and connected to the at least one WLAN; and
	[1.3]	one or more processors configured to apply a stored WWAN differential traffic control policy to Internet data service provided using the WWAN modem and the at least one WWAN, and
	[1.4]	indicate to a particular application, via an application program interface (API), one or more network access conditions based on the applied WWAN differential traffic control policy,
	[1.5]	wherein the one or more network access conditions includes a network access condition that indicates the unavailability to the particular application of an Internet data service that is currently available via the WWAN modem to a different application, but that is indicated as unavailable to the particular application.
2	[2]	The wireless end-user device of claim 1, wherein the API indicates the unavailability to the particular application of an Internet data service that is currently available by emulating network messaging.
3	[3]	The wireless end-user device of claim 1, wherein the one or more processors are further configured to apply the differential traffic control policy to the particular application when a user of the device is not directly

		interacting with the particular application or perceiving any benefit from the particular application.
4	[4]	The wireless end-user device of claim 1, further comprising a user interface to provide a user of the device with information regarding why the differential traffic control policy is applied to the particular application.
5	[5]	The wireless end-user device of claim 1, further comprising a user interface to inform a user of the device when there are options to set, control, override, or modify service usage controls that affect the differential traffic control policy.
6	[6]	The wireless end-user device of claim 1, wherein the differential traffic control policy is part of a multimode profile having different policies for different networks.
7	[7]	The wireless end-user device of claim 6, wherein the one or more processors are further configured to select a traffic control policy from the multimode profile based at least in part on the type of network connection currently in use by the device.
8	[8]	The wireless end-user device of claim 7, wherein the one or more processors are further configured to, when the type of network connection is at least one type of WLAN connection, select a traffic control policy from the multimode profile based at least in part on a type of network connection from the WLAN to the Internet.
9	[9]	The wireless end-user device of claim 1, further comprising a network stack interface integrated with the API.
10	[10.1]	The wireless end-user device of claim 1, further comprising a networking stack,
	[10.2]	wherein the one or more processors are further configured to, at an application service interface layer, identify application traffic flows prior to the flows entering the networking stack.
11	[11]	The wireless end-user device of claim 1, wherein the API comprises a network access API.

12	[12]	The wireless end-user device of claim 1, wherein the one or more network access conditions indicated via the API to the particular application comprises information on whether a current connected WWAN is a roaming network or a non-roaming network.
13	[13]	The wireless end-user device of claim 1, wherein the API informs the particular application when it is allowed to access Internet data service that is currently available via the WWAN modem.
14	[14]	The wireless end-user device of claim 1, wherein the API informs the particular application of one or more network traffic controls that the application is expected to implement.
15	[15]	The wireless end-user device of claim 1, wherein the API instructs the particular application to transition to a different state.
16	[16]	The wireless end-user device of claim 1, wherein the API instructs the device to transition to a power save state.
17	[17]	The wireless end-user device of claim 1, wherein the one or more processors are further configured to dynamically change the application of the stored WWAN differential traffic control policy based on a power state of the device.
18	[18]	The wireless end-user device of claim 1, wherein the one or more processors are further configured to dynamically change the application of the stored WWAN differential traffic control policy based on a device usage state.
19	[19]	The wireless end-user device of claim 1, wherein the one or more processors are further configured to dynamically change the application of the stored WWAN differential traffic control policy based on monitoring of user interaction with the device.
20	[20]	The wireless end-user device of claim 1, wherein the one or more processors are further configured to dynamically change the application of the stored WWAN differential traffic control policy based on power control state changes for one or more of the modems.
21	[21]	The wireless end-user device of claim 1, wherein the one or more processors are configured to associate the

		particular application with the differential traffic control policy based on an application behavior.
22	[22]	The wireless end-user device of claim 1, wherein the differential traffic control policy defines that applications to which the policy applies can only have WWAN network access events during particular time windows.
23	[23.1]	The wireless end-user device of claim 1, further comprising a network stack interface that intercepts network socket interface messages for applications and OS functions,
	[23.2]	the one or more processors configured to apply the stored WWAN differential traffic control policy to at least some of the intercepted network socket interface messages.
24	[24]	The wireless end-user device of claim 1, wherein the one or more processors are further configured to update the stored WWAN differential traffic control policy based on information received from a network element.
25	[25.1]	The wireless end-user device of claim 1, the WWAN differential traffic control policy comprising first and second differential traffic control policies applicable respectively to Internet data service provided using the WWAN modem to connect to a home WWAN and a roaming WWAN,
	[25.2]	wherein the one or more processors are further configured to apply the first differential traffic control policy when Internet data service is provided through a home WWAN and to apply the second differential traffic control policy when Internet data service is provided through a roaming WWAN.
26	[26]	The wireless end-user device of claim 1, wherein the one or more processors are further configured to determine that the different application is an active application based on a user interaction with the different application.
27	[27]	The wireless end-user device of claim 1, further comprising an agent to block, modify, remove, or replace user interface messages generated by the particular application based on the applied WWAN differential traffic control policy.

28	[28]	The wireless end-user device of claim 1, wherein the one or more processors are configured to apply the stored WWAN differential traffic control policy to selectively block network access by the particular application by intercepting open, connect, and/or write requests by the particular application to a network stack.
29	[29]	The wireless end-user device of claim 2, wherein emulating network messaging comprises responding to a network request from the particular application by blocking the request from passing to a network stack and returning to the particular application a message indicating the network request was not successful.

C. Prosecution History

41. Application Serial No. 14/673,625 was filed with 1 original claim, which was replaced in a preliminary amendment. SAMSUNG-1002, 691-98, 873. The Examiner allowed these claims without issuing any Office Action. *Id.*, 122-28. The Examiner also never considered Shell, Araujo, and Montemurro or advanced any substantive rejection based on these references. *See* SAMSUNG-1001, Cover.

IX. INTERPRETATION OF THE '359 PATENT CLAIMS

42. I understand that, for purposes of my analysis in this *inter partes* review proceeding, the terms appearing in a patent claim should be interpreted according to their “ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. § 42.100(b). In that regard, I understand that the best indicator of claim meaning is its usage in the context of the patent specification as

understood by a POSITA. I further understand that the words of the claims should be given their plain meaning unless that meaning is inconsistent with the patent specification or the patent's history of examination before the Patent Office. I also understand that the words of the claims should be interpreted as they would have been interpreted by a POSITA at the time of the invention was made (not today). Because I do not know at what date the invention as claimed was made, if ever, I have used the Critical Date as the point in time for claim interpretation purposes. My opinion does not change if the invention date is earlier.

X. SUMMARY OF RELEVANT PRIOR ART

A. Shell

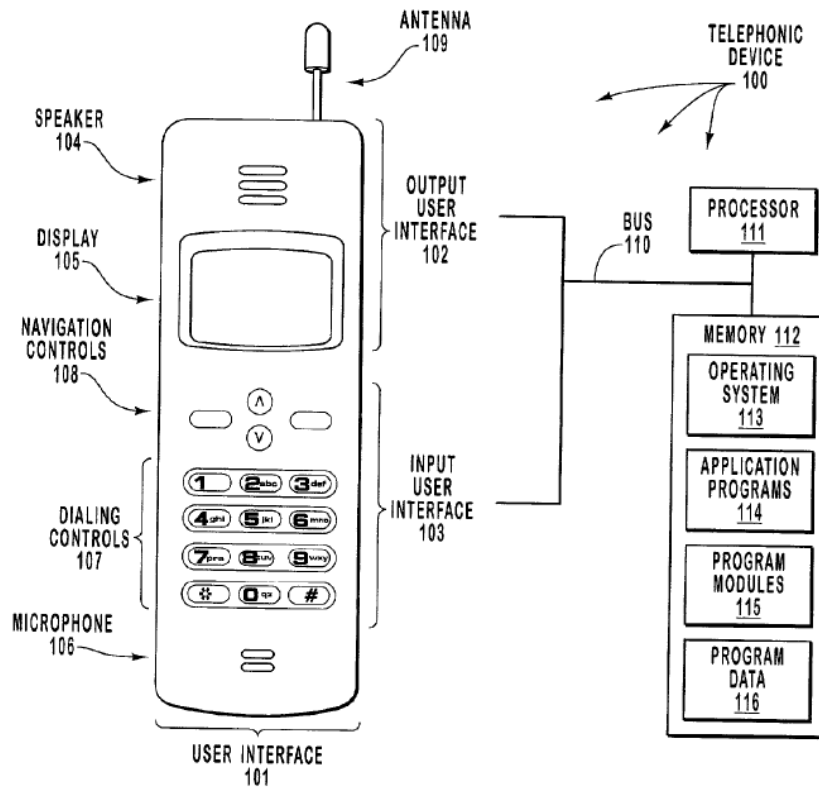
43. Shell is generally focused on “[i]ncreasing the level of automation when establishing and managing network connections.” SAMSUNG-1005, Abstract. This is accomplished using “connection manager” modules that “relieve[] users from having to know detailed information about [a] remote resource and [a] destination network.” *Id.* Specifically, “[w]hen the connection manager receives a request to access a remote resource,” it uses “extensible decision logic to identify a most appropriate connection method for connecting to a destination network.” *Id.* Through its connection manager modules, the Shell system “abstract[s] connection information from a user so as to limit required user interactions when creating network connections.” *Id.*, [0003].

44. Shell recognizes that “conditions associated with a network may make establishment of a connection using a particular connection method undesirable at certain times,” e.g., if a network is experiencing “heavy usage” or “high degree of latency.” SAMSUNG-1005, [0007]. This is because a user may not be “aware of the costs associated with different types of connections” and thus may use a “less desirable connection method to access a resource.” *Id.* Beyond this issue, Shell attributes “[o]ther problems associated with connection methods [that] also exist at the programming level.” *Id.*, [0009]. Because “[m]ost connection methods utilize different Application Program Interfaces (‘APIs’) when establishing a connection[,]” they may require different APIs or “different parameters,” which “[r]equires programmers to know the nuances of different connection methods” and thereby may be “time consuming” and lead to reduced efficiency. *Id.* Shell thereby focuses on “automating the establishment and management of various kinds of network connections in a manner that reduces the amount of information a user must know to establish a connection.” *Id.*, [0011]. In this way, “[a] user does not need to identify the underlying properties of the network where the resource is located or a connection method for getting to that network.” *Id.*, [0012].

45. During operation, “an application included in a computing device may request access to a remote resource located in a destination network.” SAMSUNG-1005, [0033]. A connection manager (or “connection planner”)

“analyzes a variety of factors to determine the most appropriate connection method to connect to the destination network.” *Id.* Examples of factors include “local resources currently available on the computing device, the priority of the request, or connection parameters (such as latency, cost, and bandwidth) associated with a connection method.” *Id.* “When the most appropriate connection method is known, a connection service provider associated with the connection manager may cause a system level [API] to create a connection from a local computing device to the destination network.” *Id.*

46. Figure 1 shows an example of a “suitable computing environment” for the techniques disclosed throughout Shell. SAMSUNG-1005, [0037].



SAMSUNG-1005, FIG. 1

The operating environment includes a “telephonic device 100” as a “general-purpose computing device” with standard computing elements for “complex and flexible general-purpose processing capabilities.” SAMSUNG-1005, [0039]; *see id.*, [0038]-[0044].

47. Figure 2 shows an “arrangement” of “functional components” that “offers a level of abstraction between application modules that may request a connection, and the system level APIs that actually establish the connections.” SAMSUNG-1005, [0045].

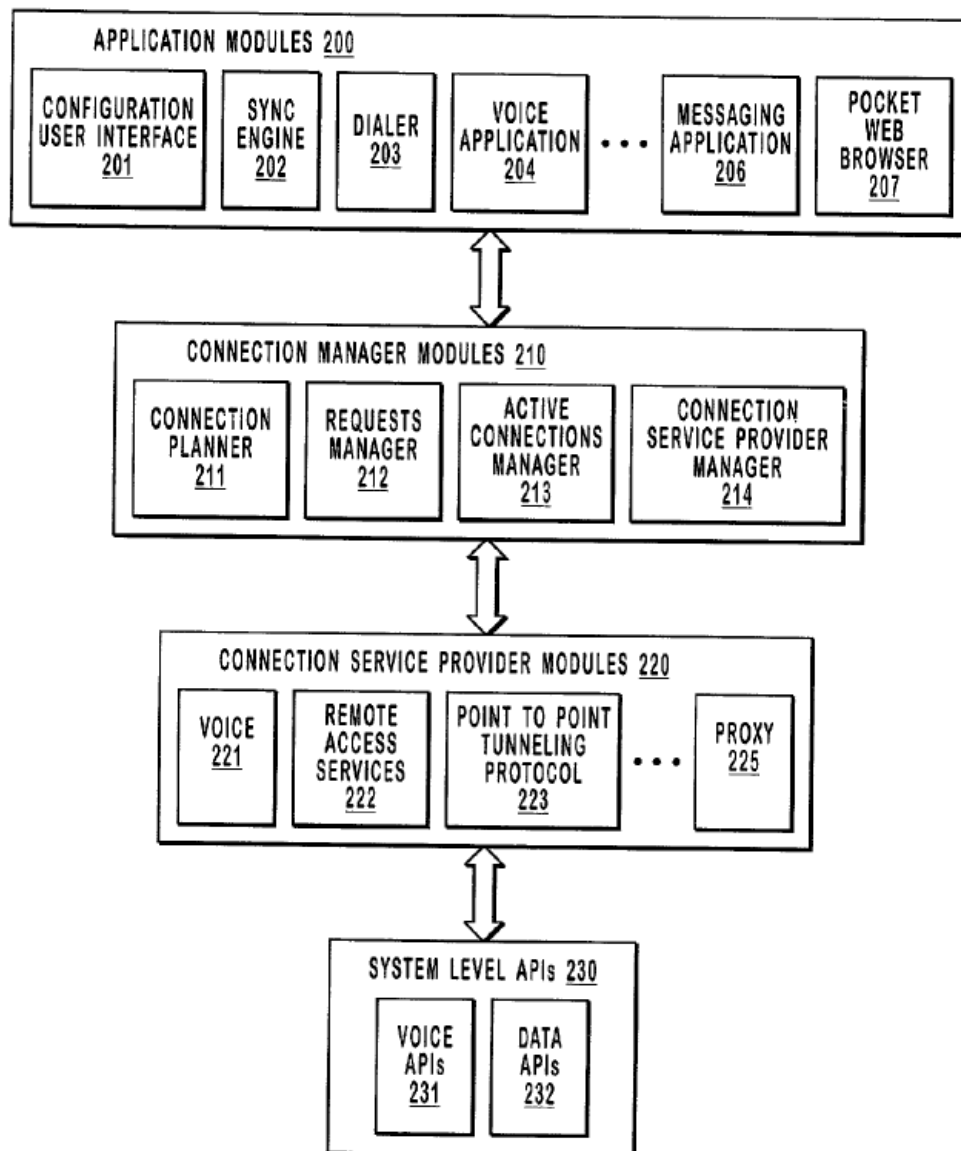


FIG. 2

SAMSUNG-1005, FIG. 2

Specifically, “[c]onnection manager modules 210 and connection service provider modules 220 may be configured as APIs that receive standard data when being called by another module.” SAMSUNG-1005, [0046]. These modules “include

computer-executable instructions that cause, directly or indirectly, the appropriate calls to be sent to system level APIs 230[,]" thereby simplifying the process of establishing connections for application modules 200. *Id.* Further, "APIs associated with connection manager modules 210 and connection service provider modules 220 may be integrated into protocol stacks." *Id.*, [0048]. Thus, "APIs associated with connection manager modules 210 and connection service provider modules 220 may be called instead of APIs associated with conventional protocol stacks." *Id.*

48. Further, "connection manager modules 210 may cause resources to be released from and allocated to connection methods." SAMSUNG-1005, [0092]. "When this occurs, connection manager modules 210 may notify application modules 200 of such changes[,]" e.g., "application modules may be notified that a connection will be established, that a connection is currently active, that a connection is being disconnected, that no connection is currently active, that a connection to a resource is blocked for some reason..." *Id.* Connection manager modules 210 may also "provide connection specific parameters to application modules 200 including, bandwidth, latency, power consumption, cost, physical location, past connection successes, or other parameters..." *Id.*

49. Figure 5A shows an example of a technique of network connection request handling based on prioritization of network connection requests.

SAMSUNG-1005, [0057].

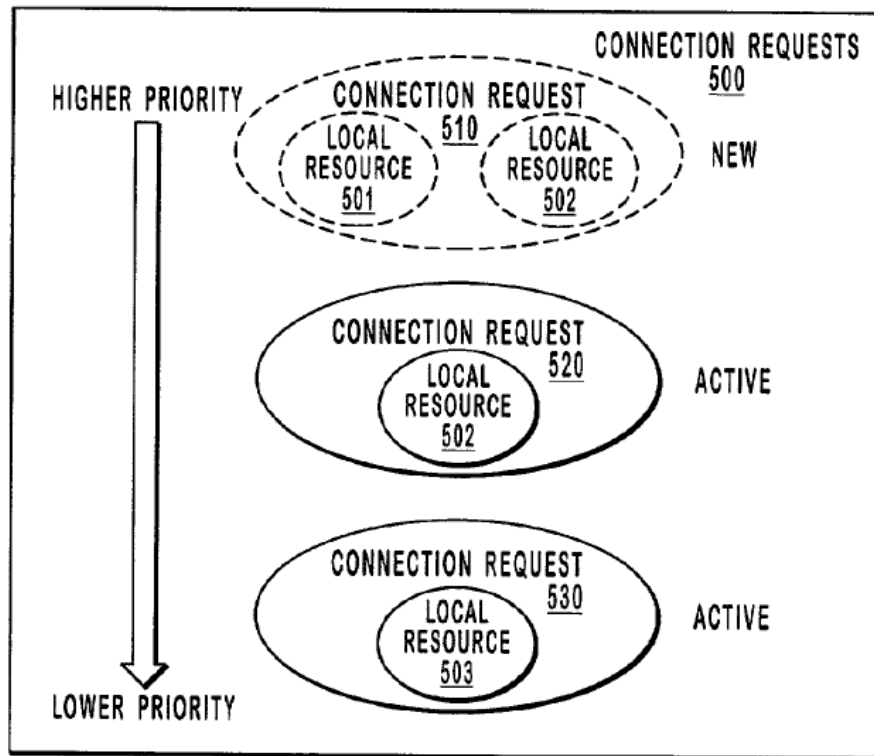


FIG. 5A

SAMSUNG-1005, FIG. 5A

In the example above, “connection requests 500 includ[e] new connection request 510 and active connection requests 520 and 530.” SAMSUNG-1005, [0057].

“Connection requests 510, 520, and 530 are prioritized, with connection 510 having the highest priority and connection 530 having the lowest priority.” *Id.* In this example, “requests from sync engine 202 may be assigned a higher priority

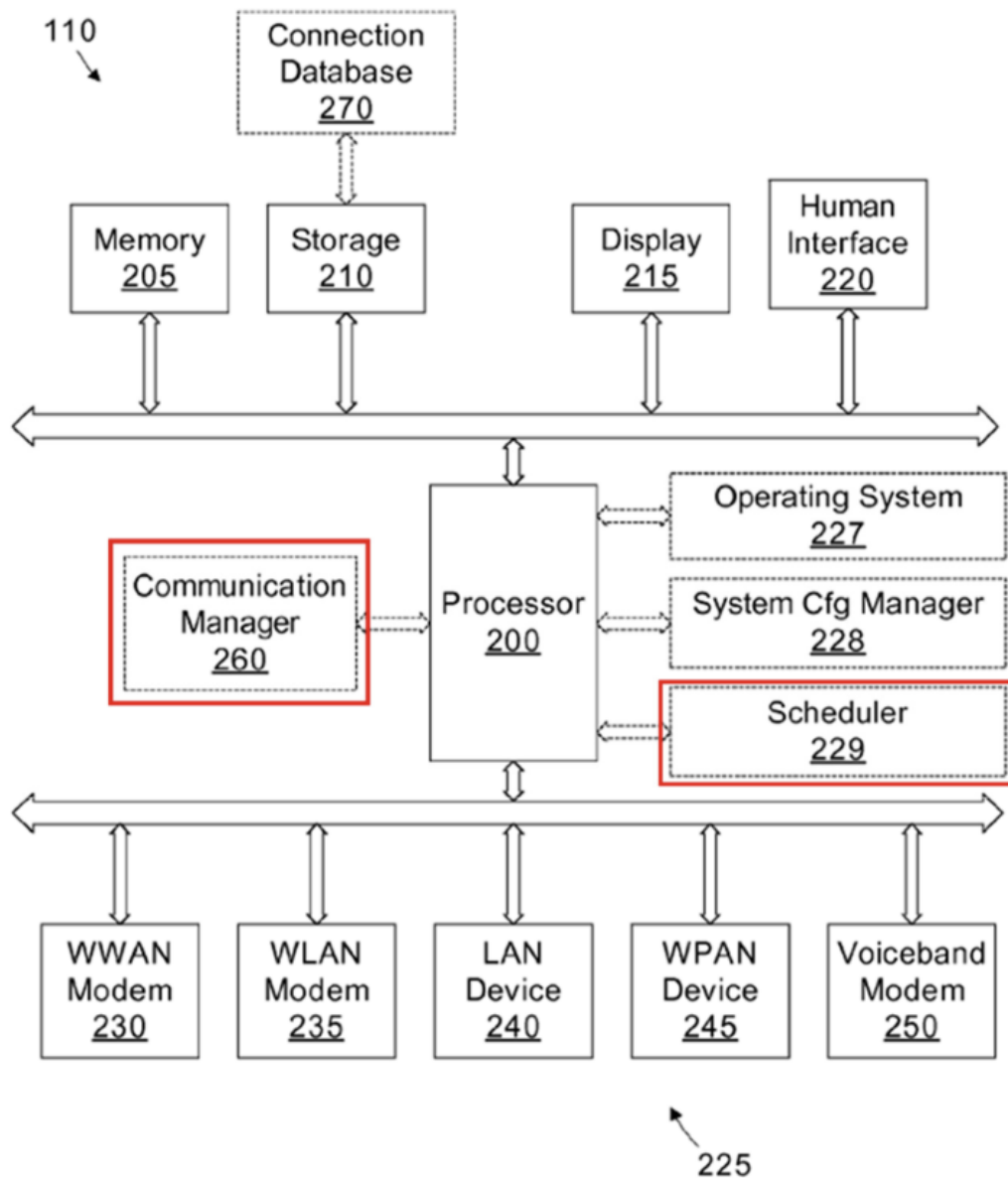
than requests from messaging application 206[,]” though “[p]riorities may also be assigned to connection requests based on the current state of applications associated with the connection requests[,]” e.g., “a request from an application that is currently active may be assigned a higher priority than a request from an application running in the background.” *Id.* Other examples of prioritization are “based on characteristics associated with the connection requests[,]”: e.g., “voice requests may be assigned a higher priority than data requests.” *Id.*

50. Shell further teaches embodiments in which “an application may notify connection [planner] 211 it wishes to ‘piggy back’ on other connection requests,” i.e., “an application may request to only access resources in a destination network when another application is already accessing resources.” SAMSUNG-1005, [0070]. In such embodiments, “[a]pplications that wish to piggyback, may call APIs associated with connection manager modules 210 and be given an open handle.” *Id.* The open handle “facilitates notification back to the application that a connection on which the application may want to piggy back has been established.” *Id.*

B. Cole

51. Cole describes “[a] method for configuring a mobile device including a plurality of communication interfaces” including a “first communication interface” and multiple “second ... communication interfaces” that are “selected

for establishing a second connection to a remote network.” SAMSUNG-1006, Abstract, [0014]. Cole uses a “connection manager 260” to select “communication interfaces” for connecting to remote networks (e.g., a wireless local area network (WLAN), a wireless wide area network (WWAN), and a wireless personal area network (WPAN)). SAMSUNG-1006, [0015], [0003], [0031], [0035], [0036], FIGS. 1, 2. The connection manager can also “schedule various events” using a “scheduler” (also annotated) to “effectively use available bandwidth.” SAMSUNG-1006, [0074], [0036], [0118], [0119]. This includes taking action when “monitored loads” exceed a “predetermined threshold.” SAMSUNG-1006, [0075], [0077].



SAMSUNG-1006, FIG. 2

52. Cole’s “connection manager” manages communication interfaces to “determine which connections should be active at any given time.” SAMSUNG-1006, [0036]. This determination includes monitoring various network connectivity metrics, such as battery power, connection bandwidth, communication

capabilities, connection availability, user preferences, application requirements, and mobile device resources. *Id.*, [0036], [0080]. For instance, in some cases, the connection manager may establish a connection with a communication interface based on application requirements. *Id.*, [0069].

53. To assist in evaluating communication interfaces, the connection manager “maintains data related to each of the communication interfaces,” to include “connection name, connection technology type, a location (LOC), a configuration flag (CFG), a bandwidth (BW), a connection time (CT), a user interaction required (UI) flag, a fast connect availability flag (FC), [] power characteristics (PC), thermal characteristics (TC), a vendor name, a technical support contact (TSC) information, an ignore flag,” and more. SAMSUNG-1006, [0037]. Cole describes that these parameters may be “based on historical data[,] ... user specified parameters, or static parameters,” and may also be “monitored” by the connection manager. SAMSUNG-1006, [0038].

54. Cole describes that, when “the monitored load exceeds a predetermined threshold (e.g., 85% of capacity), the connection manager 260 may take one of several actions to manage the load.” SAMSUNG-1006, [0075]. For example, Cole describes that applications can be “aborted,” “postpone[d],” cancelled and rescheduled, and “block[ed].” *Id.*, [0080], [0085]-[0090], [0125]. Cole additionally describes that these above actions can be taken in response to

mobile device states, including a “power state.” *Id.*, [0090]-[0098]. “By monitoring the load on an active connection, the connection manager 260 tries to proactively prevent increased latency from an overloaded connection from impacting the performance of the running applications or the efficiency of the user.” *Id.*, [0075].

C. Araujo

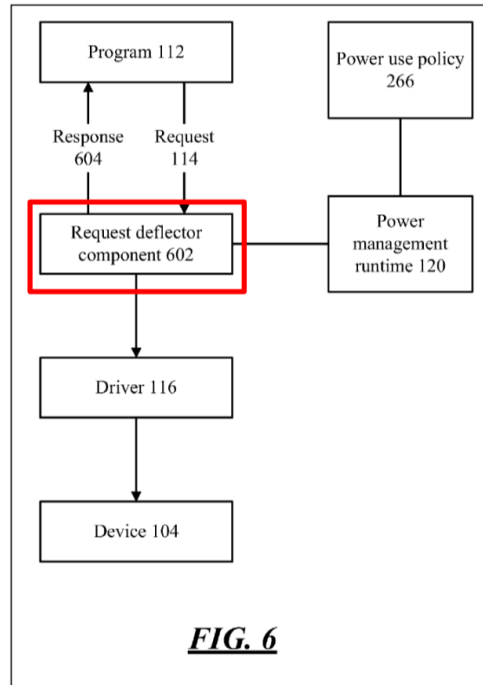
55. Araujo describes “a power-management policy” for managing a “machine's power usage.” SAMSUNG-1008, Abstract. The policy can be implemented on “[e]lectronic devices,” such as “computers, wireless telephones, audio/video equipment, etc.” *Id.*, [0014]. The policy is applied to “justify [a program] consuming power, depending on what the program is doing.” *Id.*, [0004]. By implementing this policy, a device can better “use the energy in a way that strikes a balance between providing functionality and maintaining longevity of the charge.” *Id.*, [0001].

56. In Araujo, programs are classified using various criteria, and selectively provided with resources based on their classifications. SAMSUNG-1008, [0027]. A program’s ability to consume power is “based on a finding as to whether the program's status justifies the consumption of power under the circumstances that are present.” *Id.* For example, a “very important person” (VIP)

status allows a program to “consume power in situations where other programs would not be permitted to consume power.” *Id.*

57. Further, a device can provide resources to the applications differently, based on whether the applications are “worthy” to consume power. SAMSUNG-1008, [0024]. A program can generate “a request to perform an action using a particular device, such as “a request to send data over a wireless network.” *Id.*, [0020], [0024]. The device can then perform various responsive operations, including allowing the request (SAMSUNG-1008, [0035]; FIG. 3, 302) and blocking the request (*id.*, [0035], [0050]; FIG. 4, 406).

58. The responsive operations are provided by a “request deflector component” that is “interposed” between a “program” and a “driver” of a “device,” and configured to “intercept[] requests and respond[] to the requests based on power management considerations before the request reach[es] a device driver.” SAMSUNG-1008, [0024], [0050]. Figure 6 (below) shows an example of the “request deflector component”:



SAMSUNG-1008, FIG. 6 (annotated)

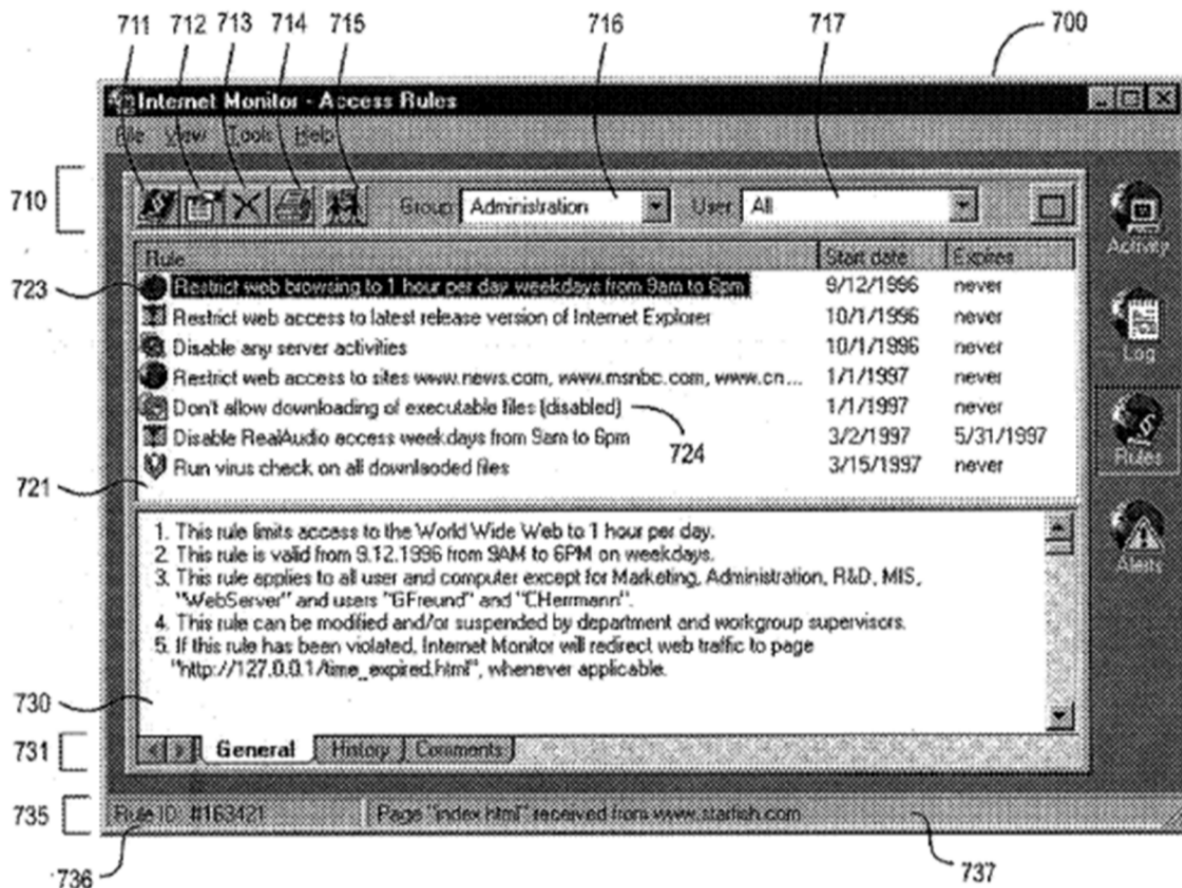
D. Freund

59. Freund relates to using “access rules” for efficient use of computing resources and controlling the use of networks. SAMSUNG-1012, 3:62-67.

“[A]ccess rules can include criteria such as total time a user can be connected to the Internet,” “time a user can interactively use the Internet,” and “a list of applications or application versions that a user can or cannot use in order to access the Internet,” among others. *Id.*, 4:8-19. To apply the rules, Freund classifies whether an application is interacting with the user in the device UI foreground. *Id.*, 10:16-20. Freund addresses challenges, including “hording[sic] available network

bandwidth through use of bandwidth-intensive applications (e.g., realtime audio programs).” *Id.*, 2:12-14.

60. To enable rule configuration, “a preferred user interface or ‘wizard’ dialogs for configuring rules,” displays “rules governing operation of the Internet access monitor” and rules that “are available for a current configuration,” and allows a user to create, edit, or delete rules. *Id.*, 24:16-24. Figure 7A (below) shows an example UI panel.



SAMSUNG-1012, FIG. 7A

Policies are loaded from a network element (“supervisor”), and “the Supervisor filters rules appropriate for the client ... and transmits them to the Client Monitor.” SAMSUNG-1012, 28:14-18. The Client Monitor “saves a copy of the rules onto a local hard disk” where they are ready for use. SAMSUNG-1012, 28:18-19. Freund relates to using “access rules” for efficient use of computing resources and controlling the use of networks. SAMSUNG-1012, 3:62-67. “[A]ccess rules can include criteria such as total time a user can be connected to the Internet,” “time a user can interactively use the Internet,” and “a list of applications or application versions that a user can or cannot use in order to access the Internet,” among others. *Id.*, 4:8-19. To apply the rules, Freund classifies whether an application is interacting with the user in the device UI foreground. *Id.*, 10:16-20. Freund addresses challenges, including “hording[sic] available network bandwidth through use of bandwidth-intensive applications (e.g., realtime audio programs).” *Id.*, 2:12-14.

E. Montemurro

61. Montemurro is focused on “policy-based routing of communications among two or more modes of wireless communication.” SAMSUNG-1009, [0001]. Devices with such connectivity capabilities are “dual or multi-mode devices,” with “radio access technologies that provide access to multiple network types,” including WLAN, WWAN. *Id.*, [0003]. But “[t]here are costs associated

with application access from these different network[s,]” and thus, “[i]t is therefore desirable to have a mechanism that seeks to optimize communications.” *Id.*, [0004].

62. Montemurro’s mechanism relies on policies to “configure connections and routes.” SAMSUNG-1009, [0011]. The is accomplished using a “rules engine” that “evaluates [the] policies on a status change...to configure a routing table.” *Id.* Together, the rules engine and the routing table “provides an appropriate connection to an application for its respective communications.” *Id.* Figure 2 (below) shows “policy-based data routing for multi-mode operations of device 102 for communicating with network infrastructure.” SAMSUNG-1009, [0024].

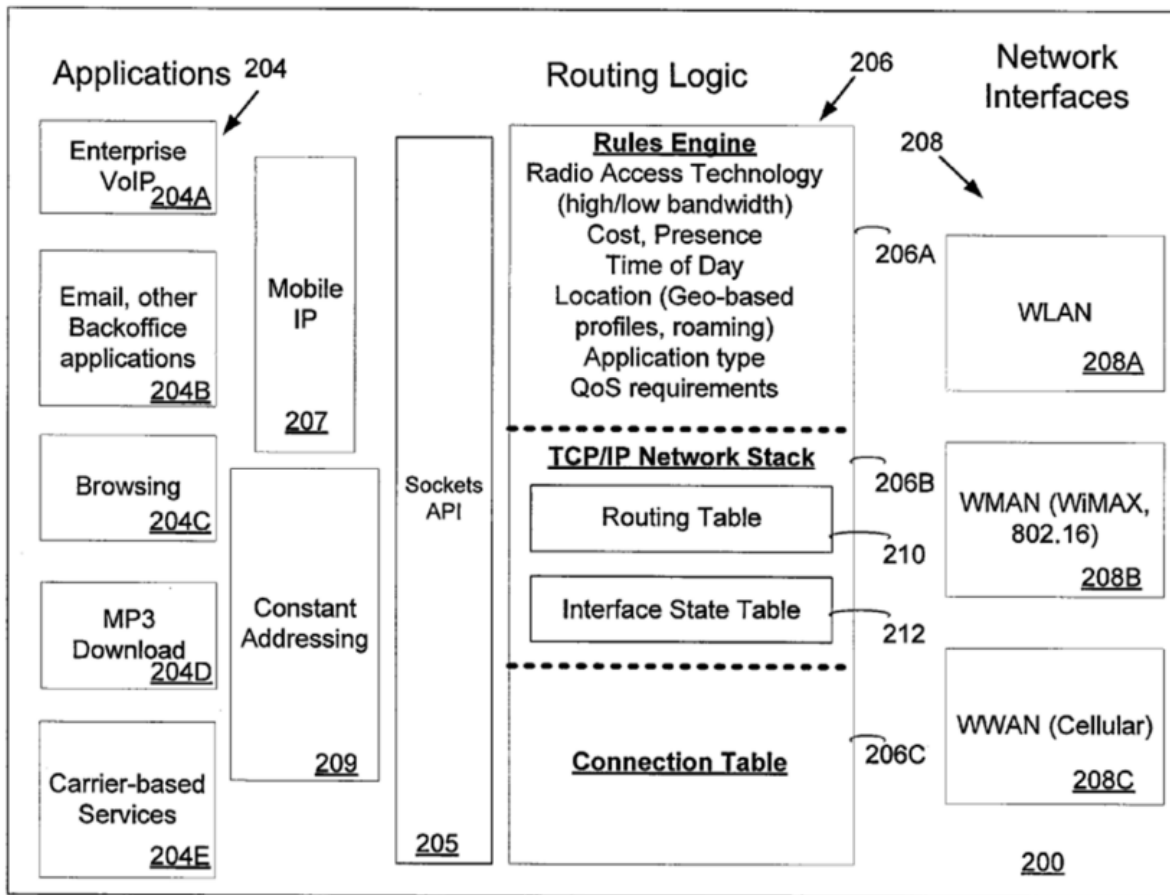


Figure 2

SAMSUNG-1009, FIG. 2

Device 102 includes “a plurality of applications 204, routing logic 206, and network interfaces 208.” SAMSUNG-1009, [0025]. Routing logic 206 includes rules engine 206A as a “policy-based mechanism using rules to configure TCP/IP network stack 206B and connection table 206C to coordinate communications for applications 204 using communications interfaces [208].” *Id.*, [0027]. For example, “a rule is a specific policy that is entered in the ‘rules engine’ for carrying out the policy.” *Id.*

63. Network interfaces 208 include interfaces for “WLAN 208A, WMAN 208B and WWAN 208C network communications,” and optionally, “short-range wireless interfaces” and “interface(s) for wired network communications.” SAMSUNG-1009, [0028]. Rules engine 206A “configures the communication operations with a set of rules/policies” for various factors, e.g., “radio access technology...cost, presence, time of day, location..., destination IP address, application type, and Quality of Service (QoS) requirements.” *Id.*, [0029].

64. Further, rules engine 206A “configures (i.e. modifies, periodically in response to changes of state and the evaluation of its rules) connection table 207C and routing table 210 to optimize the flow of communications over multiple communications modes (e.g., interfaces 208 and respective networks 104 and 106).” SAMSUNG-1009, [0033]. On a state change, e.g., “time of day,” “connecting/disconnecting of device 102 with a specific network 104 and 106,” rules engine 206A executes and modifies “routing table 210 to ensure that data goes out to the most appropriate network (via respective network interface 208).” *Id.*, [0035]. Rules engine 206A further “determine[s] which interface would be best to service a particular application.” *Id.*

XI. ANALYSIS

A. GROUND 1A: Shell-Cole Renders Claims 1, 4, 9-11, 13, 14, 18, 19, 21, 23, And 26-28 Obvious

1. Combination

65. As described above, Shell recognizes that “conditions associated with a network may make establishment of a connection using a particular connection method undesirable at certain times,” e.g., if a network is experiencing “heavy usage” or “high degree of latency.” SAMSUNG-1005, [0007]. Given such inefficiencies, Shell describes decision logic for handling connection requests from application modules, including prioritization mechanisms, addressing one challenge. *Id.*, [0057], [0092].

66. While Shell’s prioritization techniques would have improved the efficient utilization of network resources, a POSITA would have understood that, in mobile devices, power management is also critical, at least because applications cannot execute without sufficient power. A POSITA would have also recognized that network transmissions from a mobile device can cause high power utilization, especially using some types of power-intensive network connections (e.g., WWAN). SAMSUNG-1006, [0011]. Thus, when a high rate of network connection requests are provided by applications, power demand can increase, highlighting the importance of power management. SAMSUNG-1006, [0011], [0078].

67. By the Critical Date, one well-known technique for mobile device power management was selectively adjusting between available network connections (e.g., WWAN, WLAN). For example, around 2006, Lee

(SAMSUNG-1021) described how so-called vertical handoffs, which occur when a connection is switched from a WLAN to a WWAN network. SAMSUNG-1021, 1. Specifically, “[i]n particular, since most mobile terminals are battery powered, it is a challenge to design new technologies that allow mobile terminals to maintain their active connection as they move across different types of wireless networks, that is known as ‘vertical handoff’, while minimizing their power consumption”, and create algorithms for minimizing power consumption across WLAN and WWAN interfaces. SAMSUNG-1021, 1; *see* SAMSUNG-1021. In addition to reducing power consumption, this technique could also increase available bandwidth, reducing network congestion. *Id.*; SAMSUNG-1006, [0078]. For example, WLANs typically provide more bandwidth and involve greater cost relative to that involved with use of WWANs (and further involves less power usage). Given these considerations, which Shell does not specifically address, a POSITA would have been motivated to look to solutions beyond Shell to improve implementation of its techniques. *Id.*

68. To address these concerns (e.g., cost, power, network selection, and the relationship between them), a POSITA would have looked to references such as Cole, which, like Shell, are directed to providing network connectivity for mobile devices. SAMSUNG-1006, Abstract, [0014]-[0015]. As described above, Cole offers Internet access control techniques that address excess power

consumption and selection of an appropriate network. SAMSUNG-1006, [0078], [0088]. For example, Cole describes an Internet access control where the connection manager “detects a power state change (e.g., alarm or mode change),” and “[i]f the mode change is associated with an increased power state (e.g., connection to AC power or change to high performance mode), the connection manager 260 establishes the connection normally using the ‘best’ connection technique.” SAMSUNG-1006, [0098]. In another example, Cole explains that the connection manager “detects a sustained high load on the current communication link over the active communication interface,” and if load management is enabled, the connection manager “determines if there is a [communication interface] with better connectivity present” and “determines the availability of the selected device ... and attempts to connect using the selected device.” *Id.*, [0083]. In both examples, Cole determines whether to apply an Internet access control (connecting using the “best” connection technique) based on a condition (power state change of the device and high load on current communication link from device usage). SAMSUNG-1006, [0083], [0098].

69. A POSITA would have found obvious integration of Cole’s techniques with Shell’s techniques to promote more efficient communications and network access connectivity. In operation, the resulting Shell-Cole device first evaluates network and power conditions using Cole’s techniques to determine

whether to apply an Internet access control to connect to the network with the “best” connection type, then evaluates further conditions (e.g., whether an application is running in the foreground or background, which application is running, etc.) using Shell’s techniques to determine the priority of the packets traveling over the network selected by the Shell-Cole device using Cole’s techniques. *Id.*; SAMSUNG-1005, [0092]; SAMSUNG-1006, [0011], [0078], [0083], [0088], [0098].

70. In an example of combining Shell and Cole, a POSITA would have incorporated functionality of Cole’s communication manager (communication manager 260) into Shell’s connection manager modules (connection manager modules 210). The Shell-Cole device would have included a network interface containing WWAN Modem 230 (supporting 2G and 3G cellular technologies) and WLAN Modem 235 (supporting Wi-Fi) from Cole. *Id.*; SAMSUNG-1005, FIG 1; SAMSUNG-1006, [0031]. Using such connections, the Shell-Cole device would establish and manage “various kinds of network connections.” SAMSUNG-1005, [0032].

71. The Shell-Cole device would have further included elements customarily found in a mobile device, such as a processor (e.g., Shell’s CPU 111, Cole’s processor 200), non-transitory memory (e.g., Shell’s memory 112, Cole’s

memory 205), and a display (e.g., Shell's display 105, Cole's display 215).

SAMSUNG-1005, FIG. 1, [0038]-[0044]; SAMSUNG-1006, FIG. 2, [0034].

72. It would have been obvious to a POSITA that the processor of the Shell-Cole device would have been configured to perform the operations discussed in Ground 1A because it “may allow for complex and flexible general-purpose processing capabilities.” SAMSUNG-1006, [0040]. A POSITA would have further known that the processor processes instructions fetched from the main memory unit.

73. A POSITA would have expected success in incorporating power management functionality (as taught by Cole) into the connection manager modules taught by Shell. Such a combination would have required only routine programming knowledge well within the skill of a POSITA before the Critical Date. The change would have amounted to nothing more than the use of a known technique to improve similar devices – in each instance a mobile device equipped with network interfaces – in a similar way and combining prior art elements according to known methods to yield the predictable results described above. *Id.* For example, the devices of Shell and Cole include interfaces between a processor that executes the described functions and network devices that provide communication functions. SAMSUNG-1005, [0036], [0040]; SAMSUNG-1006, FIG. 2, [0034]-[0035]. A POSITA would have known that such processors

commonly run processes such as those run by the software modules described in Shell and Cole. SAMSUNG-1005, [0040]; SAMSUNG-1006, [0036]. Therefore, the combination of the techniques of Cole into Shell in the Shell-Cole device would require only routine programming knowledge.

74. The elements of the resulting Shell-Cole device would each perform functions they had been known to perform prior to the combination—communication manager modules would perform the same functions to prioritize network connection requests through use of decision logic (as taught in Shell), leveraging functionality offered by policies of Cole’s connection manager 260, including policies associated with applications, to identify the best power consumption configuration and available connection for transmission. Accordingly, a POSITA would have naturally expected success when incorporating Cole’s teachings into the Shell system, as described above. Id.

2. *Analysis*

[1.pre] A wireless end-user device, comprising:

75. If the limiting, Shell-Cole teaches it or renders it obvious. Shell’s Figure 1 depicts a telephonic device 100 (***wireless end-user device***):

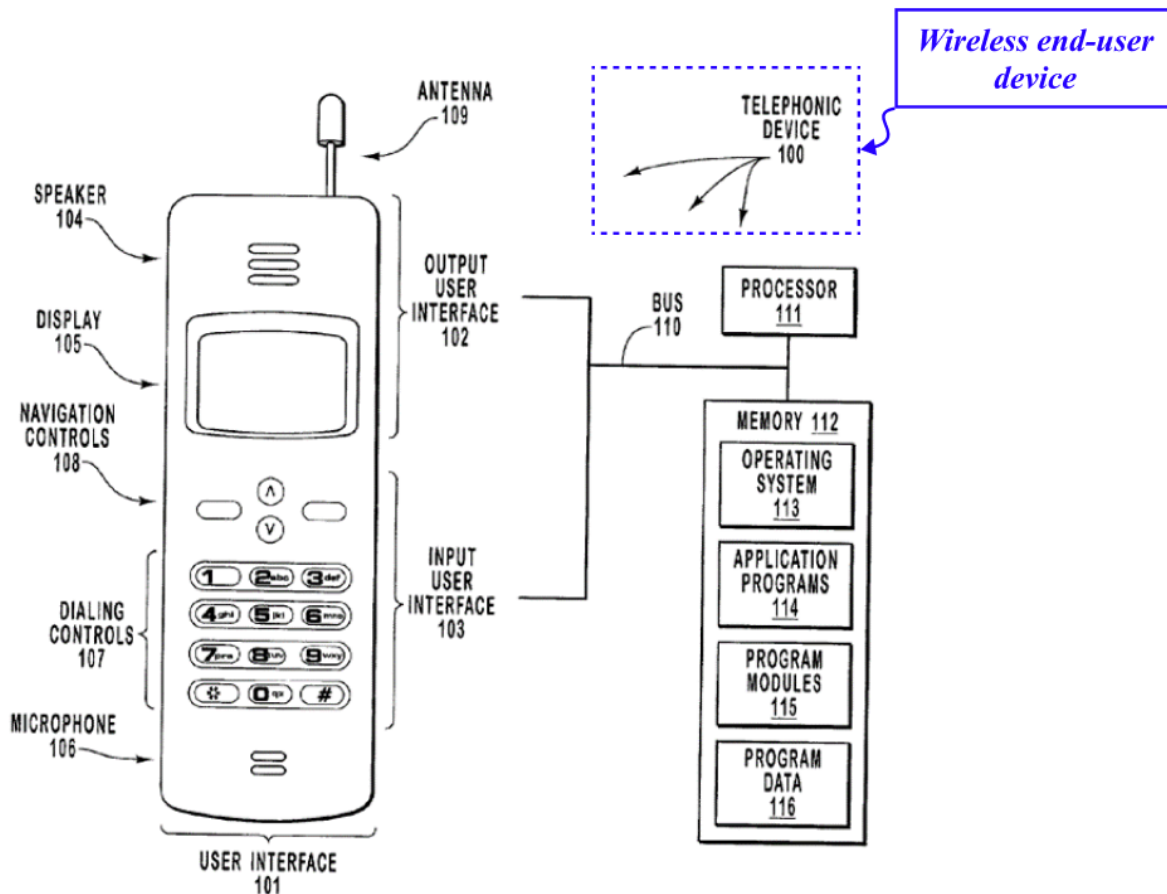


FIG. 1

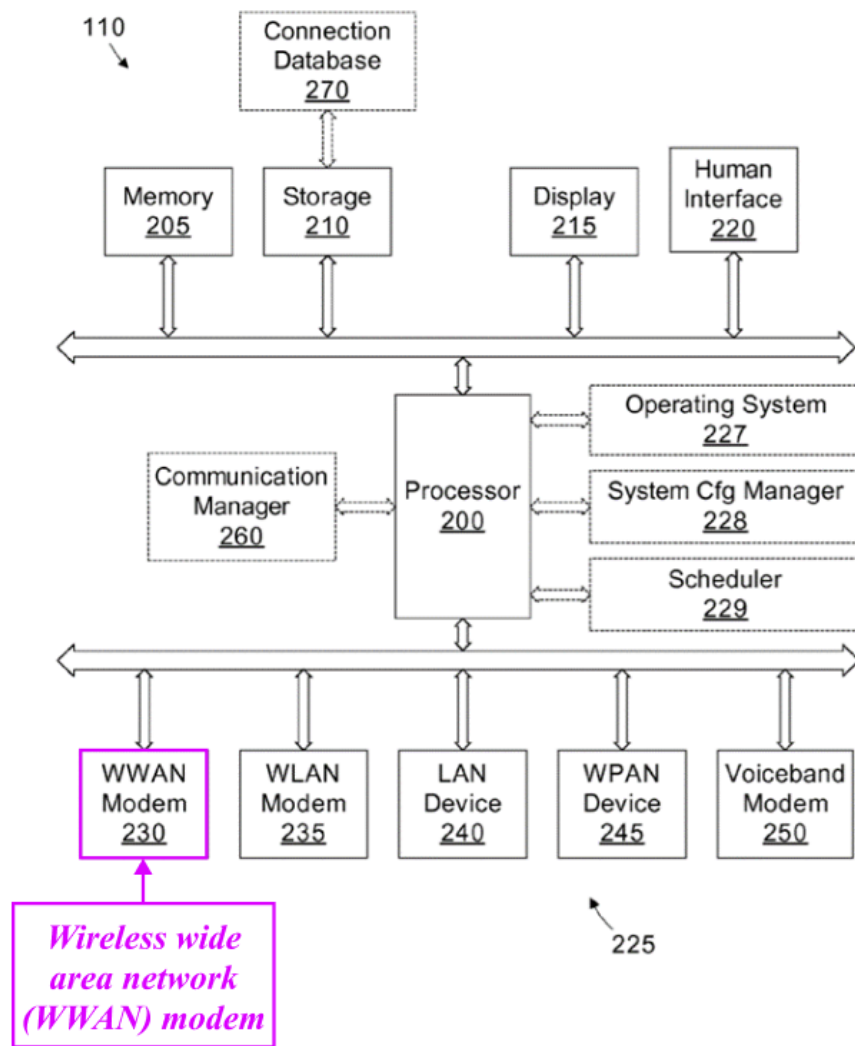
SAMSUNG-1005, FIG. 1 (annotated)

76. The telephonic device 100 can be any device running a “suitable operating environment for the principles of the invention,” including “a general-purpose computing device.” SAMSUNG-1005, [0038]. Further, the telephonic device 100 may be implemented in “many types of computer system configurations,” such “personal computers, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, pagers, and

the like.” *Id.*, [0036]. The telephonic device 100 also has “an antenna 109 if the telephone device 100 has wireless capabilities.” *Id.*, [0038].

[1.1] 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;

77. Shell-Cole renders [1.1] obvious based on disclosures in Shell and Cole regarding network connectivity. *See* SAMSUNG-1005, [0055]; SAMSUNG-1006, [0031], [0035], [0088], FIG. 2. Shell describes that “wireless carriers may define logic for accessing resources included in networks under their control[,]” suggesting connectivity of the telephonic device 100 with a cellular network (***wide area network***). SAMSUNG-1005, [0055]. Further, in Cole’s Figure 2 (below), a mobile device includes a “WWAN modem 230” (***wireless wide area network (WWAN) modem***) providing connectivity to a “wireless wide area network (WWAN) 130,” confirming that such a modem configuration was well-known by the Critical Date. SAMSUNG-1006, [0035].



SAMSUNG-1006, FIG. 2 (annotated)

78. The “WWAN modem 230” can be a “3G channel” so that a mobile device can have network connectivity when in a “public place.” SAMSUNG-1006, [0088]. Further, in Cole, a mobile device interfaces “with a plurality of communication networks 120, such as a wireless wide area network (WWAN) 130” (e.g., EVDO, HSDPA, 3G, GSM, CDMA, 2G, WiMAX, WiBro). *Id.*,

[0031]. As explained in Section X.B, a POSITA would have understood and found obvious this modem configuration described in Cole to represent implementation detail for Shell's telephonic device 100. SAMSUNG-1018, [0095], [0125] (describing a mobile device with connectivity to a "Wide Area Network (WAN) or the Internet through a variety of connections," such as "standard telephone lines, LAN or WAN links," "broadband connections," "wireless connections," or "some combination of any or all of the above"). Thus, in Shell-Cole, Shell's telephonic device incorporates Cole's WWAN modem 230 and thereby would have enabled data to be wirelessly transmitted over a WWAN network. *Id.*

79. Integration of "WWAN modem 230" from Cole into Shell's telephonic device 100 enables the Shell-Cole device to ***communicate data for Internet service activities between the device and at least one WWAN***. As described in Cole, WWAN modem 230 is designed to interface with various communication networks such as EVDO, HSDPA, 3G, GSM, CDMA, 2G, WiMAX, and WiBro. SAMSUNG-1006, [0031]. These networks provide the infrastructure necessary for high-speed data transmission and continuous connectivity. When Shell's device is configured with this modem, the device gains the ability to connect to these wide area networks, allowing the Shell-Cole device to send and receive data packets required for Internet services like web browsing, email, and streaming. *Id.* This data communication is facilitated by the modem's

capability to handle the necessary network protocols and manage the data flow between the device and the WWAN. *Id.*; SAMSUNG-1005, [0006]-[0008]; SAMSUNG-1006, [0088].

80. Moreover, Cole describes network selection techniques that further enhance the device's ability to maintain robust Internet connectivity. SAMSUNG-1006, [0006]. Specifically, WWAN modem 230 can automatically select and connect to the most appropriate WWAN based on availability and signal strength. *Id.*, [0031]. Such dynamic network selection ensures that the device remains connected to a WWAN, thereby providing continuous Internet service. The modem can function as a "3G channel" and is capable of maintaining connectivity even in public places. *Id.*, [0088]. In Shell-Cole, this automatic selection and seamless switching between networks allows the device to consistently communicate data for Internet activities, ensuring that users can access online services reliably whenever the device is within the coverage of any supported WWAN. This capability, derived from Cole's detailed network interface descriptions, illustrates how the combined disclosures of Shell and Cole render the Shell-Cole device's data communication functionality over a WWAN both feasible and obvious. *Id.*

81. Further, accessing a network resource (as described in Shell) involves the transmission and reception of data over a network, which represents examples

of an *Internet service activity* involving data exchange over the Internet. In both cases, the underlying process includes establishing a connection (for example, the creation of a network socket between the source and destination), managing data packets (for example, allowing Internet packets carried over the HSPDA protocol in 3G networks), and ensuring communication between the device and the network.

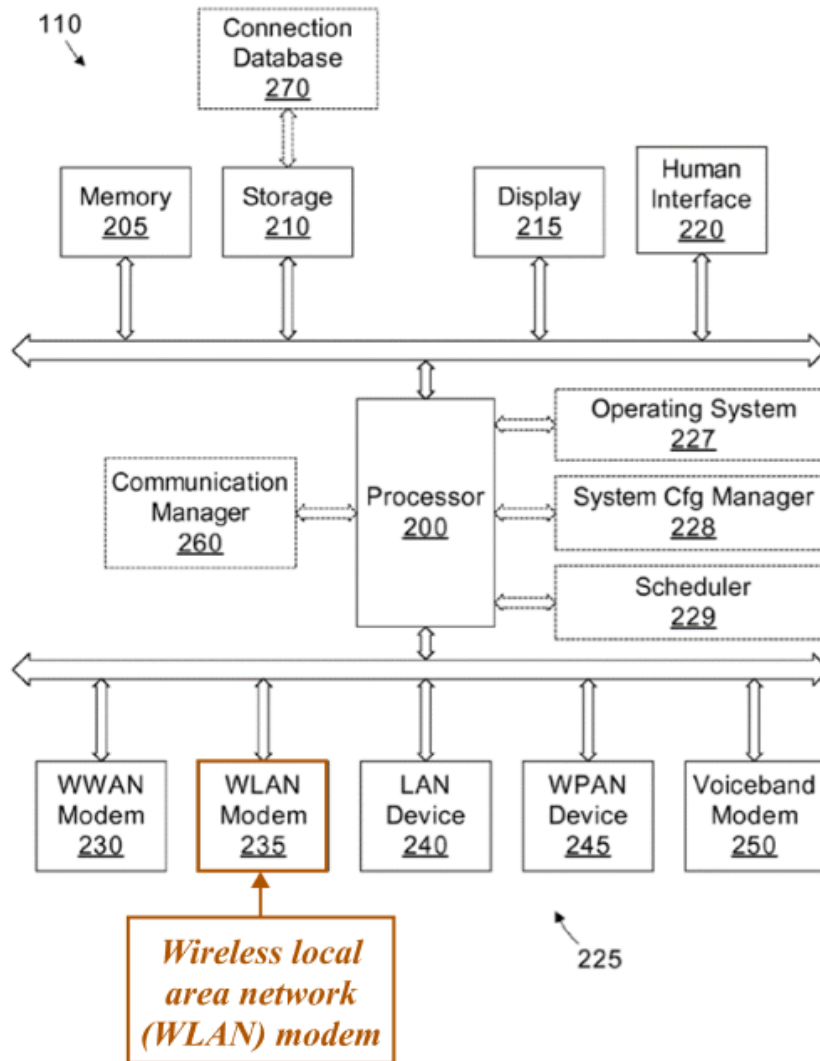
82. Additionally, the mechanisms and protocols used for accessing general network resources directly support Internet service activities, making them analogous in their operational principles and requirements. For example, Stevens (SAMSUNG-1030) discusses how network sockets can be used to create connections between a source and destination and how they can be configured to allow for activity requests. Stevens states:

When a socket is created by the *socket* function, it is assumed to be an active socket, that is, a client socket that will issue a *connect*. The *listen* function converts an unconnected socket into a passive socket, indicating that the kernel should accept incoming connection requests directed to this socket.”

SAMSUNG-1030, 9.

[1.2] a wireless local area network (WLAN) modem to receive and transmit Internet data between the device and at least one WLAN when configured for and connected to the at least one WLAN; and

83. Shell-Cole renders [1.2] obvious based on disclosures in Shell and Cole regarding network connectivity. *See* SAMSUNG-1005, [0035]; SAMSUNG-1006, [0031], [0088], FIG. 2. In Cole’s Figure 2 (below), a mobile device includes a “WLAN modem 235” (***wireless local area network (WLAN) modem***) providing connectivity to a “wireless local area network (WLAN) 140,” confirming that such a modem configuration was well-known by the Critical Date. SAMSUNG-1006, [0035].



SAMSUNG-1006, FIG. 2 (annotated)

84. The “WLAN modem 235” can be a “Wi-Fi connection” so that a mobile device can have network connectivity “[w]hen the user returns home.” SAMSUNG-1006, [0088]. Further, in Cole, a mobile device interfaces “with a plurality of communication networks 120,” including a “wireless local area network (WLAN) 140” (e.g., IEEE 802.11, Wi-Fi). *Id.*, [0031]. As explained in Section XI.A.1, a POSITA would have understood and found obvious this modem

configuration described in Cole to represent implementation detail for Shell's telephonic device 100. *See* SAMSUNG-1018, [0095], [0125] (describing a mobile device with connectivity to a "Wide Area Network (WAN) or the Internet through a variety of connections," such as "standard telephone lines, LAN or WAN links," "broadband connections," "wireless connections," or "some combination of any or all of the above"). Thus, in Shell-Cole, Shell's telephonic device incorporates Cole's WLAN modem 235 and thereby would have enabled data to be wirelessly transmitted over a WLAN network. *Id.*

85. Integration of "WLAN modem 235" from Cole into Shell's telephonic device 100 enables the Shell-Cole device to ***communicate data for Internet service activities between the device and at least WLAN***. As described in Cole, WLAN modem 235 is designed to interface with wireless local area networks, such as IEEE 802.11 and Wi-Fi. SAMSUNG-1006, [0031]. These networks provide the infrastructure necessary for high-speed data transmission and continuous connectivity while reducing network bandwidth costs. When Shell's device is configured with this modem, the device gains the ability to connect to these local area networks, allowing the Shell-Cole device to send and receive data packets required for Internet services like web browsing, email, and streaming, without requiring access to a wide area network. *Id.* This data communication is facilitated by the modem's capability to handle the necessary network protocols

and manage the data flow between the device and the WLAN. *Id.*; SAMSUNG-1005, [0006]-[0008]; SAMSUNG-1006, [0088].

86. Moreover, as discussed for [1.pre], Cole describes network selection techniques that further enhance the device's ability to maintain robust Internet connectivity. SAMSUNG-1006, [0006]. Such dynamic network selection ensures that the device remains connected to a WLAN, thereby providing continuous Internet service. The modem can provide connectivity to a "home wireless network" or a "office wireless network" and is capable of maintaining connectivity with lower bandwidth costs. *Id.*, [0038]. In Shell-Cole, this automatic selection and seamless switching between networks allows the device to consistently communicate data for Internet activities, ensuring that users can access online services reliably whenever the device is within the coverage of any supported WLAN. *Id.*, [0004] ("[t]he wide variety of connection options provides the user with flexibility and the ability to connect to a network in virtually any location"). This capability, derived from Cole's detailed network interface descriptions, illustrates how the combined disclosures of Shell and Cole render the Shell-Cole device's data communication functionality over a WLAN both feasible and obvious. *Id.* Further, as discussed for [1.pre], accessing a network resource (as described in Shell) involves the transmission and reception of data over a

network, which represents examples of an *Internet service activity* involving data exchange over the Internet.

[1.3] one or more processors configured to apply a stored WWAN differential traffic control policy to Internet data service provided using the WWAN modem and the at least one WWAN,

87. Shell-Cole discloses [1.3] based on the processing capabilities of a mobile device taught in Shell. *See* SAMSUNG-1005, [0040], FIG. 1; SAMSUNG-1006, [0034]. In Shell, telephonic device 100 includes a processor 111 (***one or more processors***) with “complex and flexible general-purpose processing capabilities.” SAMSUNG-1005, [0040]. Cole likewise describes a mobile device 110 with a processor 200. SAMSUNG-1006, [0015], [0034], [0091]; *see also* Section XI.A.2 (discussing that Shell-Cole’s device would have further included elements customarily found in a mobile device, such as a processor (e.g., Shell’s CPU 111, Cole’s processor 200)).

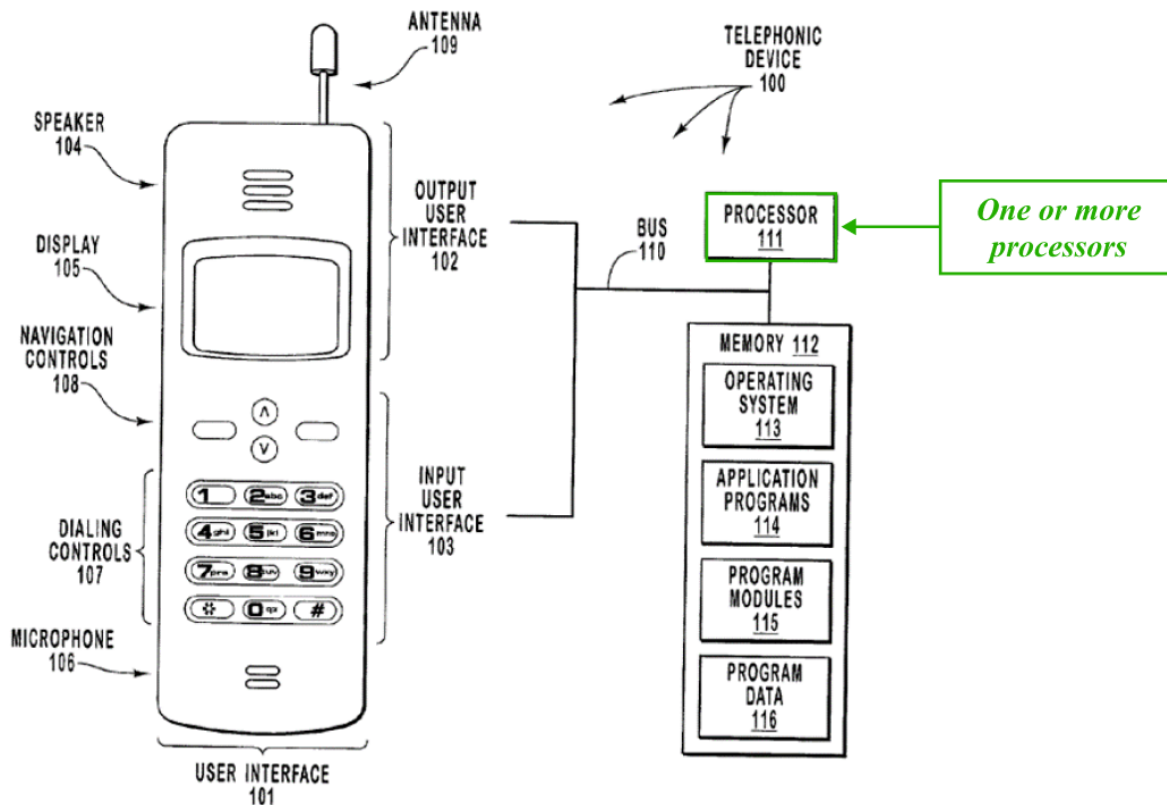


FIG. 1

SAMSUNG-1005, FIG. 1 (annotated)

88. Shell-Cole also renders obvious application of a *stored [] differential traffic control policy to Internet data service* based on Shell's disclosures of "connection manager modules" regulating network access by applications on telephonic device 100. See SAMSUNG-1005, [0013], [0017], [0032]; SAMSUNG-1006, [0009], [0080], [0085]. As discussed in Section XI.A.1, in Shell, the "connection manager modules" are configured to regulate network access by applications, e.g., by prioritizing connection requests, determining an

appropriate method for establishing a network connection, granting a connection access request, etc. *See* SAMSUNG-1005, [0013], [0017], [0032].

89. Specifically, a POSITA would have recognized that network connection request handling (*Internet data service*) is performed by Shell's "connection manager modules" and that such functionality encompasses examples of applying a *traffic control policy* described in the '359 patent, such as "block/allow settings," "throttle settings," "adaptive throttle settings," and "QoS class settings." SAMSUNG-1001, 85:59-86:24. Further, as noted in the '359 patent, the "traffic control policy" can be used for "selecting the network from a list of available networks, blocking or reducing access until a connection is made to an alternative network, and/or modifying or replacing a network stack interface of the device to provide for intercept or discontinuance of network socket interface messages to applications or OS functions." *Id.*

90. Like these examples, Shell teaches "connection manager modules" (or a "connection planner") receiving a connection request provided by an application that "request[s] access to a remote resource located in a destination network." SAMSUNG-1005, [0033]. The connection manager modules then "analyze[] a variety of factors to determine the most appropriate connection method to connect to the destination network." *Id.* Further, "[w]hen the most appropriate connection method is known, a connection service provider associated with the connection

manager may cause a system level Application Programmer Interface ('API') to create a connection from a local computing device to the destination network.” *Id.* By granting or refusing connection requests, the connection planner is configured to apply decision logic (*stored [] differential traffic control policy*) for network connection request handling (*Internet data service*) (e.g., “block/allow settings”). *See* SAMSUNG-1001, 85:59-86:24.

91. As another example, Shell describes that “connection requests may be prioritized so that higher priority requests cause lower priority requests to release a connection method so it may be used by the higher priority request.” SAMSUNG-1005, [0013]. Further, “[p]riorities may be assigned to connection requests based on applications associated with the connection requests.” *Id.*, [0057]. This is shown in Figure 5A below:

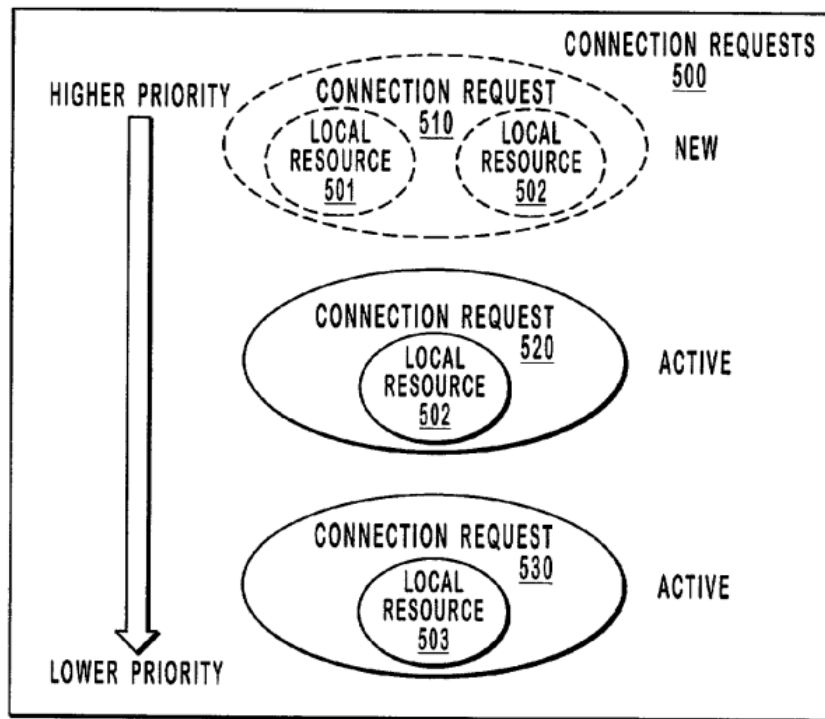


FIG. 5A

SAMSUNG-1005, FIG. 5A

In the example shown above, “[c]onnection requests 510, 520, and 530 are prioritized, with connection 510 having the highest priority and connection 530 having the lowest priority.” *Id.*, [0057]. A POSITA would have recognized that Shell’s prioritization techniques of network connection requests coincides with ’359 patent specification’s disclosure of “blocking or reducing access until a connection is made to an alternative network” as an example of applying a “traffic control policy.” SAMSUNG-1001, 85:59-86:24.

92. Further, a POSITA would have found it obvious that decision logic applied by the Shell-Cole device would have provided a WWAN differential

traffic control policy based on disclosures in Shell and Cole regarding regulating network access in relation to WWAN networks. For example, Shell teaches multiple types of applications, e.g., data application(s), voice application(s), running on telephonic device 100 at the same time and submitting connection requests to an available network, such as a cellular network (e.g., WWAN) provided by a “wireless carrier.” SAMSUNG-1005, [0055]. In this example, when a voice application submits a request to connect to the cellular network, the connection manager modules, for example, may prioritize the voice application over the data application. *See id.*, [0013] (“a voice call may be given higher [] [*sic*] priority over data calls that share the same modem”). When this occurs, the data application module receives a notification that its “connection” is “being disconnected....” *Id.*, [0092]. A POSITA would have thereby recognized that decision logic applied in this example relates to access to a cellular network and thereby involves applying a ***WWAN differential traffic control policy***.

93. Further, Cole similarly describes a “connection manager 260” that may perform “communication load monitoring and deferred action functions,” which are “useful in many situations.” SAMSUNG-1006, [0088]. For example, if “the user is surfing a web site and selects a file for download[,]” “[t]he file [may be] larger than the user anticipated and is taking a long time to transfer.” *Id.* In this example, “the lengthy transfer is conducted using the WWAN modem 230

(e.g., a 3G channel), which consumes significant battery power[,]” and thus, “[t]he connection manager 260 identifies the load on the WWAN modem 230 and asks the user to find a better connection.” *Id.* As another example, where “a scheduled synchronization starts while the user is traveling[,]” the connection manager 260 may perform traffic monitoring to reduce “consum[ing] battery power.”

SAMSUNG-1006, [0089]. The connection manager 260 “notices the load from the synchronization operation and asks the user if it should be deferred (i.e., no better connection is available).” *Id.*

[1.4] indicate to a particular application, via an application program interface (API), one or more network access conditions based on the applied WWAN differential traffic control policy,

94. Shell-Cole renders [1.4] obvious through Shell’s disclosures of network access handling through improved connection manager modules. *See* SAMSUNG-1005, [0013], [0048], [0055], [0092]. As explained below, the improved connection manager modules provide notification data (*indicates*) to an application module (*particular application*) representing decision logic regarding a network connection request (*one or more network access conditions*). Further, a POSITA would have known or found obvious that the Shell-Cole device implements the decision logic using one or more APIs (*application program interface (API)*) through the application of network connection handling (*applied [] differential traffic control policy*).

95. As discussed for [1.3], Shell-Cole includes Cole's WWAN modem 230, which would have made it obvious that decision logic applied by the Shell-Cole device, in some circumstances, involves network connection handling relating to connection requests to access resources associated with a WWAN network (***WWAN differential traffic control policy***). SAMSUNG-1005, [0013], [0055], [0092]; SAMSUNG-1006, [0015], [0034], [0091].

96. Specifically, Shell describes that "application modules may be notified" of a decision regarding a network connection request, and the "connection manager" may also "provide connection specific parameters" to the application modules. SAMSUNG-1005, [0092]. Thus, a POSITA would have recognized that, in Shell-Cole, processor 111 of telephonic device 100 is configured to provide notification data (e.g., decision, connection-specific parameters) related to a network connection request (***indicate***) to an application module that provided the network connection request (***particular application***).

97. The notification data also includes decision logic ***one or more network access conditions***. SAMSUNG-1005, [0092]. For example, the notification data may relate to a decision regarding the network connection request, e.g., "a connection will be established," "a connection is currently active," "connection is being disconnected," "no connection is currently active," "a connection to a resource is blocked for some reason," or any other notification

associated with the connection to resources” (each of which would have been understood to represent *one or more network access conditions*). *Id.* As another example, the notification data may include “connection specific parameters,” e.g., “bandwidth, latency, power consumption, cost, physical location, past connection success, or other parameters” (each of which would have similarly been understood to represent *one or more network access conditions*). *Id.*

98. Further, in Shell-Cole, the connection manager modules are associated with “APIs...[that] may be called...” SAMSUNG-1005, [0048]. Specifically, the “connection manager operates between system level APIs and application programs, providing a level of abstraction.” *Id.*, Abstract; *see id.*, [0009], [0022], [0045]-[0047]; *see also id.*, [0048] (“APIs associated with connection manager modules 210”), [0070] (“[t]he open handle facilitates notification back to the application”). Based at least on these disclosures, a POSITA would have found it obvious that, in some implementations, functionality of the connection manager modules of Shell-Cole (e.g., providing notification data to application modules) may be implemented through use of well-known techniques for “providing a level of abstraction” between system-level APIs and applications. SAMSUNG-1005, [0048]. As one example, Irwin (SAMSUNG-1013), which was published before the Critical Date on 8/7/2008, describes a Widget Engine 11 (with a connection manager 16) running on a mobile device and configured to call APIs relating to

connection requests. For example, Irwin describes using APIs for communication of network-related information to an application.

99. Specifically, Irwin discusses an example of a widget that calls a stock ticker function every hour that calls a function. “The function sends a request to a specific URL, parses the response and displays the latest share price to the user. When the timer fires, the resultant HTTP request will trigger the setup of a PDP context, thereby enabling the request and response to be sent at least partially over a wireless network.” See Appx Z, [0033]. Figure 2 the following lines of code within the `getStockPrice()` function,

```
var req = new XMLHttpRequest();  
req.open(http://-----URL-----);  
req.send();  
var xml=req.responseXML;
```

These lines of code represent the use of an API in order to enact the functionality discussed above. Namely, a data structure of type `XMLHttpRequest()` is created and assigned the variable name “req”. This object req can be used for further processing based on the API, with the “`req.open()`” command creating a connection to a remote Internet destination and the “`req.send()`” apparently causing the request to be sent. A response from the destination is then collected into a variable “xml” through the API call `req.responseXML`. As these commands are not ones that

would be exposed by an operating system, and do not represent functionality implemented directly within an application, they instead represent API calls associated with the creation and use of a network connection.

100. Additionally, before the Critical Date, mobile operating systems (e.g., Symbian OS) were known to include connection management services that used APIs to allow applications to access network connectivity information. *See* SAMSUNG-1014, 45-46. For example, the Symbian OS was known to use an API for providing network information to applications. Another example is SAMSUNG-1010 (Dive-Reclus), which described a mobile device configuration that uses permissions regulating network access by a third-party application using an API call. SAMSUNG-1010, 16:23-30, 17:1-28. A POSITA would have understood that a permission decision (e.g., grant, deny) is communicated to the third-party application through a corresponding API. A POSITA would have found it obvious that, in Shell-Cole, the connection manager modules would have been configured to provide notification data to application modules through use of well-known APIs used to provide network-related information (*via an application program interface (API)*).

101. Communicating network decisions to another application was a well-known means of providing information about authorization to access resources. There were many groups that attempted to provide application programming

interfaces to access network functionality. Moerdijk (SAMSUNG-1027) described these efforts in 2003, writing that “different groups like Parlay [3] and JAIN [4,5] continued with efforts to develop application programming interfaces (APIs) that willow applications access to network functionality.” SAMSUNG-1027, 1.

Moerdijk further provides an overview of an API specification, including an API for “connectivity management.” SAMSUNG-1027, 3.

102. Further, use of an API to communicate with an application was a well-known technique prior to the '359 patent and would have been within a POSITA's general knowledge. In my opinion, the use of an API does not make the '359 patent's claims unique from prior art as of the Critical Date, and a POSITA would have found it obvious to use an API to communicate Shell's network access conditions (e.g., connection request decisions and connection specific parameters) to applications. Additionally, this is particularly applicable when an application's connection request is blocked from accessing a network resource because the application is running in the background (e.g., not interacting in the device display foreground with the user and needs some means of receiving an indication to adjust its operation).

[1.5] wherein the one or more network access conditions includes a network access condition that indicates the unavailability to the particular application of an Internet data service that is currently available via the

WWAN modem to a different application, but that is indicated as unavailable to the particular application.

103. As discussed for [1.4], notification data provided by the connection manager modules of Shell-Cole include decision logic representing ***one or more network access conditions*** (e.g., a decision regarding the network connection request). See SAMSUNG-1005, [0048], [0070], [0092].

104. Shell further describes examples in which an application may “wish[] to ‘piggy back’ on other connection requests[,]” i.e., “an application may request to only access resources in a destination network when another application is already accessing resources.” SAMSUNG-1005, [0070]. A POSITA would have recognized and/or found obvious that a “piggy back” connection request may be denied in some instances, e.g., due to limited network bandwidth. See *id.*, [0092] (“application modules may be notified that...a connection to a resource is blocked for some reason). In these instances, a first application (***the particular application***) sends a “piggy back” network connection request based on an established connection of a second application (***different application***). In denying/blocking the “piggy back” network connection request, the connection manager modules would provide notification data to the first application indicating decision information regarding the “piggy back” network connection request (***network access condition***). Further, the decision information would indicate that

the “piggy back” network connection request is denied/blocked (*Internet data service* being *indicated as unavailable to the particular application*), but that network access by the first application is available (*Internet data service [] is currently available via the WWAN modem to a different application*). *Id.*

105. APIs that allowed applications to learn of the unavailability of a connection were well known at the time of the Critical Date. The Java socket API discussed by Flanagan (SAMSUNG-1011) provides an example of this in code as a client application that connects to a server. A code comment describes functionality as “Create a socket to communicate with a server on port 6789 of the host that the applet’s code is on. Create streams that use the socket. Then create a TextField for user input and a TextArea for server output. Finally, create a thread to wait for and display server output.” SAMSUNG-1011, 3. A fragment of the code follows:

```
try {  
    s = new Socket(this.getCodeBase().getHost(), PORT);  
    in = new DataInputStream(s.getInputStream());  
    out = new PrintStream(s.getOutputStream());  
    ...  
}  
catch (IOException e) this.showStatus(e.toString());
```

Within this code block, there is an API call to create a socket and some input and output functionality with the connected socket. If the socket fails to be established,

which could occur because of connectivity issues such as a blocked request, an error is reported back to the application in the form of an IOException. The details of the error are provided in text through the showStatus function.

[4] The wireless end-user device of claim 1, further comprising a user interface to provide a user of the device with information regarding why the differential traffic control policy is applied to the particular application.

106. Shell-Cole would have incorporated the functionality to output a user interface relating to various aspects of policies used for prioritization (“user interface”). Shell describes that “configuration user interfaces” may be used to provide connection request handling techniques described within its disclosure. *See, e.g.,* SAMSUNG-1005, [0044], [0049]. For example, “configuration user interface 201 may be used to configure connection methods” and allow service personnel to “configure connection methods in a manner that is suitable to entities that have an interest in connection methods.” *Id.*, [0049]. A POSITA would have found it obvious to configure the connection manager modules in Shell-Cole to similarly provide a configuration user interface (***user interface***) to provide information regarding various aspects of connection request handling, including information regarding why decision logic is applied to application modules (***information regarding why the first differential traffic control policy is applied to the first end-user application***).

[9] The wireless end-user device of claim 1, further comprising a network stack interface integrated with the API.

107. As discussed for [1.4], in Shell-Cole, connection manager modules are associated with “APIs...[that] may be called...” (SAMSUNG-1005, [0048]) and would have operated “between system level APIs and application programs, providing a level of abstraction.” (*id.*, Abstract). See SAMSUNG-1005, [0009], [0022], [0045]-[0047]; *see also id.*, [0048] (“APIs associated with connection manager modules 210”), [0070] (“[t]he open handle facilitates notification back to the application”). Based at least on these disclosures, a POSITA would have found it obvious that, in some implementations, functionality of the connection manager modules (e.g., providing notification data to application modules) may be implemented through use of well-known API-based techniques. See SAMSUNG-1013, [0013], [0023], FIG. 2. For example, a POSITA would have known that APIs were used for communication of network-related information to an application.

108. For example, the Java socket API will communicate issues with the establishment of network connectivity to the application. As discussed above, Flanagan (SAMSUNG-1011) provides a code sample demonstrating that the Java socket API will report an error message to the application if a connection cannot be established from a source to the destination.

109. The communication manager modules of Shell-Cole are integrated into a “protocol stack” that provide connectivity to different types of network connections. SAMSUNG-1005, [0048], [0061], [0083], [0091]. For example, “a protocol stack may include multiple interfaces to system level APIs 230[,]” and “[e]ach interface to system level APIs may be associated with a separate connection method[,]” which “lead[s] to an associated computing device including multiple network addresses.” *Id.*, [0091]. This is shown below in Shell’s Figure 2:

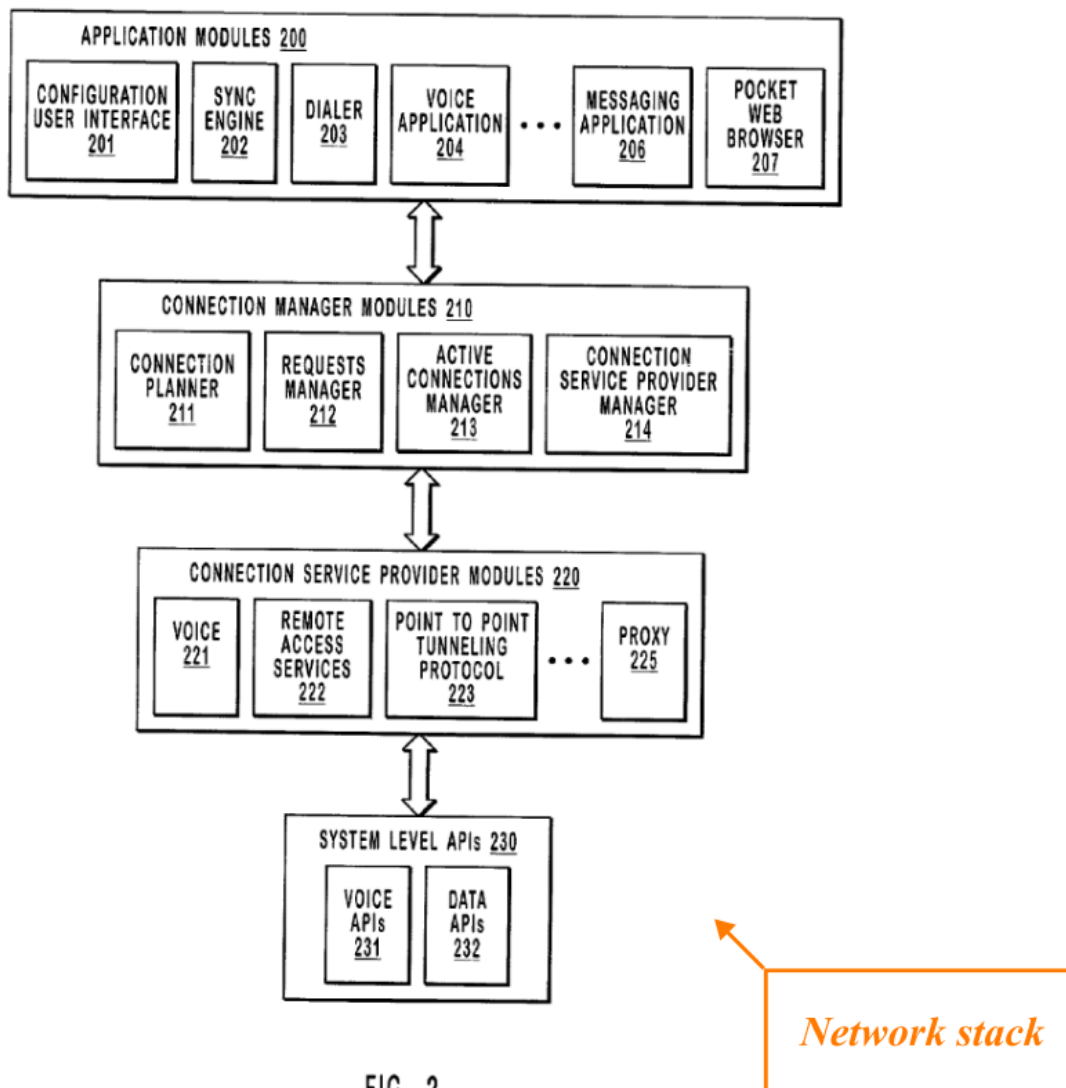


FIG. 2

SAMSUNG-1005, FIG. 2 (annotated)

Based on this disclosure and a POSITA's general knowledge of network stack and API implementation details, a POSITA would have understood and found obvious that the connection manager modules of Shell-Cole are part of the protocol stack (*network stack*) and configured to interact with several APIs, including system level APIs associated with different network connections and APIs used to

communicate network information to application modules (*interface integrated with the API*), or at least would have found such an implementation obvious based on Shell's disclosure and the well-known nature of APIs and network stacks as common programming techniques that the '359 patent did not invent.

SAMSUNG-1005, Abstract, [0048]; *see* SAMSUNG-1013, [0013], [0023], FIG. 2; SAMSUNG-1014, 45-56; SAMSUNG-1010, 16:23-30, 17:1-28.

[10.1] The wireless end-user device of claim 1, further comprising a networking stack,

110. As described for [9], a POSITA would have understood and found obvious that the connection manager modules of Shell-Cole are part of the protocol stack (*networking stack*). SAMSUNG-1005, Abstract, [0048], [0061], [0083], [0091]; *see* SAMSUNG-1013, [0013], [0023], FIG. 2; SAMSUNG-1014, 45-56; SAMSUNG-1010, 16:23-30, 17:1-28. A protocol stack is a structured framework that defines how data is transmitted and received across a network by organizing protocols in layers, each serving a specific function. This layered approach enables interoperability and simplifies the development and troubleshooting of network communications, making it a fundamental example of a networking stack.

[10.2] wherein the one or more processors are further configured to, at an application service interface layer, identify application traffic flows prior to the flows entering the networking stack.

111. As discussed for [10.1], a POSITA would have understood and found obvious that the connection manager modules of Shell-Cole are part of the protocol

stack (*networking stack*). SAMSUNG-1005, Abstract, [0048], [0061], [0083], [0091]; *see* SAMSUNG-1013, [0013], [0023], FIG. 2; SAMSUNG-1014, 45-56; SAMSUNG-1010, 16:23-30, 17:1-28.

112. A POSITA would have understood and found obvious that the protocol stack of Shell-Cole includes an *application service interface layer* because, as described in Shell, connection manager modules “operate[] between system level APIs and application programs, providing a level of abstraction.”

SAMSUNG-1005, Abstract. This is shown in Shell’s Figure 2:

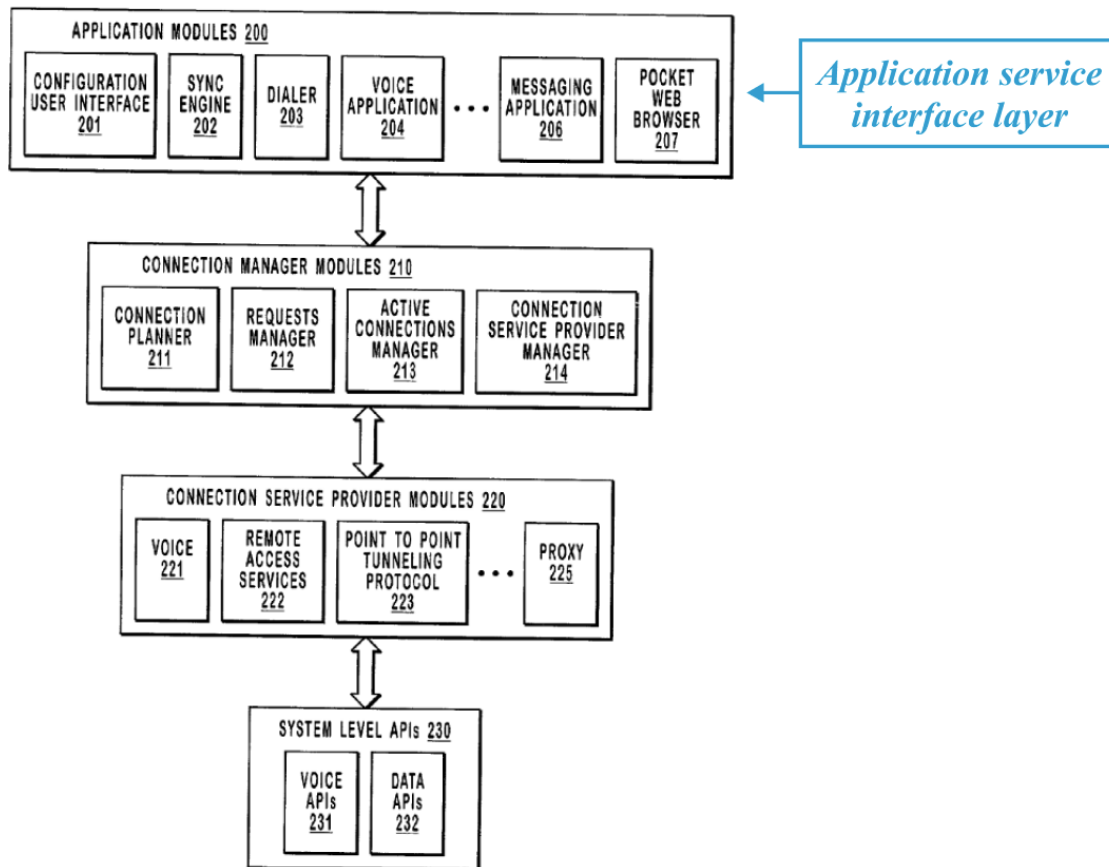


FIG. 2

SAMSUNG-1005, FIG. 2 (annotated)

113. As described in the '359 patent specification, examples of the “application service interface layer” include “QoS application service interface layer” and “network capacity controlled services interface layer.” SAMSUNG-1001, 60:33-61:6. Further, the function of the “application service interface layer” is “to monitor and in some cases interpret and process the traffic between the applications and the standard networking stack API,” e.g., by “identif[ying] application traffic flows before the application traffic flows are more difficult or practically impossible to identify farther down the stack.” *Id.* From these disclosures, a POSITA would have understood and found obvious that the protocol stack of Shell-Cole similarly includes an ***application service interface layer*** since Shell teaches how connection manager modules may block a network connection request provided by an application module to a network stack, which involves intercepting the network connection request. SAMSUNG-1005, [0092]; *see id.*, [0048], [0053], [0061], [0083], [0091].

114. For example, a software module (e.g., connection manager modules of Shell-Cole) that block network connection requests (per [0092] of Shell) would have been configured to do so at an early stage can identify application traffic flows based on the request details, such as destination address and port. By handling these requests before they are transmitted over the network, the

connection manager modules of Shell-Cole can accurately categorize the traffic, which becomes more challenging once the data is encrypted or mixed with other traffic. This early identification allows for precise traffic management and security enforcement, which is harder to achieve deeper in the network stack where data may be obscured. This functionality coincides with the '359 patent specification's description of an "application service interface layer" being configured to "identif[y] application traffic flows before the application traffic flows are more difficult or practically impossible to identify farther down the stack." SAMSUNG-1001, 60:33-61:6.

115. A POSITA would have understood and found obvious that network connection request handling, as provided by Shell-Cole, involves identifying data traffic generated by application modules because, as described in Shell, connection manager modules "analyze[] a variety of actors to determine the most appropriate connection method [to allow an application module] to connect to a destination network." SAMSUNG-1005, [0033]. Further, a "priority of [a] request," as determined by the connection manager modules may be based on information associated with application traffic data, e.g., "latency, cost, and bandwidth." *Id.* Thus, a POSITA would have also found it obvious that the Shell-Cole device is configured to *identify application traffic flows prior to flows entering the network stack* through, for example, Shell's disclosure of using "connection parameters" to

determine priority of network connection requests. SAMSUNG-1005, [0033], [0056]. Further, a POSITA would have understood and found obvious that, based on the architectures disclosed in Shell (e.g., Figure 5), connection request prioritization occurs at the “application layer,” which sits at a general protocol stack architecture.

116. Further, as explained above, a POSITA would have understood and found obvious that the protocol stack of Shell-Cole similarly includes functionality associated with the *application service interface layer*, including intercepting a network connection request (e.g., by blocking the network connection request), which similarly would have involved *identify[ing] application traffic flows* (e.g., in assessing application traffic associated with the network connection request in determining to block the network connection request). SAMSUNG-1005, [0092]; *see id.*, [0048], [0053], [0061], [0083], [0091].

[11] The wireless end-user device of claim 1, wherein the API comprises a network access API.

117. As discussed for [1.4], the connection manager of Shell-Cole would have leveraged several types of APIs (including the recited *API*) in regulating network connectivity of applications. *See* SAMSUNG-1005, [0013], [0048], [0055], [0092]; SAMSUNG-1011, 149-50. A POSITA would have understood and found obvious that APIs leveraged by Shell-Cole’s connection manager include a

network access API because the connection manager modules provide functionality focused on controlling connections to the network, and thus, relate to network access by applications. *See id.*, [0022], [0033], [0045], [0046], [0048]. For example, a POSITA would have found it obvious that the APIs leveraged by the connection manager are used to facilitate the management and control of network connections, thereby enabling or restricting access.

118. Additionally, the APIs used by the connection manager modules of Shell-Cole would have been known to interface with network protocols and services to handle connectivity, thereby making obvious that they serve the purpose of network access. For example, the UNIX socket API is an example of a well-known API used to interface with network protocols such as the Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) among others, and used for accessing network services such as the World-Wide Web, email, and file transfer. Sockets are used to connect to services and to transfer data, as described by Stevens (SAMSUNG-1030): “When a socket is created by the *socket* function, it is assumed to be an active socket, that is, a client socket that will issue a *connect*. The *listen* function converts an unconnected socket into a passive socket, indicating that the kernel should accept incoming connection requests directed to this socket.” SAMSUNG-1030, 9. The *socket*,

connect, and *listen* functions described by Stevens are all API calls to facilitate network access.

119. APIs related to providing network access information to applications were well-known before the Critical Date. For example, before the Critical Date, mobile operating systems (e.g., Symbian OS) were known to include connection management services that used APIs to allow applications to access network connectivity information. *See* SAMSUNG-1014, 45-46. As another example, Dive-Reclus (SAMSUNG-1010) teaches using permissions to regulate API calls (granting/denying) made by applications in accessing a network. SAMSUNG-1010, 16:23-17:28, 20:1-12. This aligns with the '359 patent's disclosure that a "network access API" is "used to implement traffic control for network capacity controlled services." SAMSUNG-1001, 92:1-26.

[13] The wireless end-user device of claim 1, wherein the API informs the particular application when it is allowed to access Internet data service that is currently available via the WWAN modem.

120. As discussed for [1.4], the improved connection manager of Shell-Cole would have leveraged several types of APIs (including the recited ***API***) in regulating network connectivity of applications. SAMSUNG-1005, [0013], [0048], [0055], [0092]. A POSITA would have understood and found obvious that APIs leveraged by Shell-Cole's improved connection manager includes those described in Shell, as well as conventional APIs known by the Critical Date, such

as Java sockets APIs. SAMSUNG-1011, 149-50. When a network access condition is used to permit an application to access network resources, a POSITA would have understood and found obvious that the Java sockets API provides an indication to the application of the network access condition (*the API informs the particular application when it is allowed to access Internet data service that is currently available via the WWAN modem*). See SAMSUNG-1011. This aligns with Shell's disclosure of using APIs in network connection handling.

SAMSUNG-1005, [0009], [0022], [0033], [0045]-[0048], [0070], [0083], [0090], [0091]. Further, in Shell, applications can be notified "that a connection will be established" or "that a connection is currently active," meaning that the application is notified when a connection is granted (*the API informs the particular application when it is allowed to access Internet data service*). *Id.*, [0092].

121. As explained above for [1.4], communicating network decisions to applications was a well-known means of providing information about authorization to access resources. There were many groups that attempted to provide application programming interfaces to access network functionality. Moerdijk (SAMSUNG-1027) described these efforts in 2003, writing that "different groups like Parlay [3] and JAIN [4,5] continued with efforts to develop application programming interfaces (APIs) that allow applications access to network functionality."

SAMSUNG-1027, 1. Moerdijk further provides an overview of an API specification, including an API for “connectivity management.” *Id.*

122. As another example, Dive-Reclus (SAMSUNG-1010) teaches using permissions to regulate API calls (granting/blocking) made by applications in accessing a network, and a POSITA would have found obvious that granting/blocking permission for an API call by an application involves informing the application when it is allowed to access a network service available via the network modem. SAMSUNG-1010, 16:23-17:28, 20:1-12.

[14] The wireless end-user device of claim 1, wherein the API informs the particular application of one or more network traffic controls that the application is expected to implement.

123. See [1.4] (addressing *API*); SAMSUNG-1005, [0013], [0048], [0055], [0092]; SAMSUNG-1011, 149-50. A POSITA would have understood and found obvious that APIs leveraged by Shell-Cole’s improved connection manager include those described in Shell, as well as conventional APIs known by the Critical Date, such as APIs used for providing network information to applications requesting access to network resources. See SAMSUNG-1014, 45-46 (API for providing network information to applications in the Symbian OS), SAMSUNG-1010, 16:23-30-17:1-28 (mobile device configuration using permissions to regulate network access by third-party applications using an API call); see also SAMSUNG-1011, 149-50 (Java sockets API).

124. Further, when a network access condition is used to permit an application to access network resources, a POSITA would have understood and found obvious that the Java sockets API provides instructions (e.g., opening connections, sending/receiving data, and closing connections) to the application relating to the network access condition, thereby controlling the application's access to network resources (*wherein the API informs the particular application of one or more network traffic controls that the application is expected to implement*).

125. In Shell-Cole, the connection manager modules provide “connection specific parameters,” e.g., “bandwidth, latency, power consumption, cost, physical location, past connection success,” in providing connection request handling. SAMSUNG-1005, [0092]. A POSITA would have understood and found obvious that parameters used by the connection manager modules are examples of *one or more network traffic controls* because they represent metrics used in controlling network traffic.

126. From Shell's disclosure, a POSITA would have recognized these parameters are used to optimize and manage network resources effectively. SAMSUNG-1005, [0009], [0010], [0015], [0033]. Additionally, by prioritizing or restricting certain traffic types, allocating bandwidth, , or balancing loads based on the network's current conditions and requirements. This approach ensures that

critical applications receive the necessary resources to function smoothly, minimizing latency and improving overall performance. Furthermore, it allows for dynamic adjustments to be made in real-time, accommodating fluctuations in user demand and enhancing the network's resilience against potential disruptions.

127. Shell also describes that an application module with a lower priority may be denied access when providing a network connection request, which results in the application module being provided with information of one or more network traffic controls that the application is expected to implement. SAMSUNG-1005, [0092] (“...application modules may be notified,” including that “connection is being disconnected” and “a connection to a resource is blocked for some reason”). Dive-Reclus (SAMSUNG-1010) also teaches permission decisions regulating network access by third-party applications that use API calls. SAMSUNG-1010, 16:23-17:28, 20:1-12. A POSITA would have found obvious that a permission decision is an example of *one or more network traffic controls* that controls network access, which is communicated to the application through a corresponding API (*API informs the particular application of one or more network traffic controls*).

[18] The wireless end-user device of claim 1, wherein the one or more processors are further configured to dynamically change the application

of the stored WWAN differential traffic control policy based on a device usage state.

128. Shell-Cole renders this claim obvious through Shell's disclosure of prioritizing network connections for applications. SAMSUNG-1005, [0057]. Shell explains that priorities may be "assigned to connection requests based on the current state of applications associated with the connection requests," e.g., "a request from an application that is currently active may be assigned a higher priority than a request from an application running in the background." *Id.* A POSITA would have known that application usage on a device relates to device usage, and thus, Shell's prioritization based on usage of applications providing connection requests considers whether the applications are executed in the foreground (i.e., active) or in the background. A POSITA would have known that systems routinely monitored which applications were active or inactive. *Id.* (citing SAMSUNG-1016). Thus, a POSITA would have recognized that the Shell-Cole device performs ongoing monitoring of which applications are active, so it can ***dynamically change*** the application of network connection handling (***stored WWAN differential traffic control policy***) based on the state of applications running on the device (***device usage state***). *see* SAMSUNG-1005, [0057].

129. Additionally, Shell discloses examples of *device usage*, e.g., “activating a control in a user interface, for example, a button, a drop down list or other element...,” which are monitored to initiate requests to access a remote resource. SAMSUNG-1005, [0051]. Monitored device usage is used to identify “the most appropriate connection method for connecting to the destination network (i.e. resource),” enabling the connection to be established “automatically, without requiring any user intervention at all.” *Id.*, [0052]. A POSITA would have understood and found obvious that this is yet another example of *dynamically chang[ing]* application of network connection handling (*application of the first differential traffic control policy*) based on monitoring device usage (*based on a device usage state*).

130. As another example, Shell further describes that “connection manager modules 210 may cause resources to be released from and allocated to connection methods[,]” and “[w]hen this occurs, connection manager modules 210 may notify application modules 200 of such changes.” SAMSUNG-1005, [0092]. In this example, where resource allocations for an application are “change[d]” a POSITA would have understood and found obvious that this would have similarly resulted in *dynamically chang[ing]* application of network connection handling (*application of the first differential traffic control policy*) based on monitoring

device usage (*based on a device usage state*). Additionally, “application modules may be notified” of these changes. SAMSUNG-1005, [0092].

[19] The wireless end-user device of claim 1, wherein the one or more processors are further configured to dynamically change the application of the stored WWAN differential traffic control policy based on monitoring of user interaction with the device.

131. See [18]. A POSITA would have recognized that the Shell-Cole device performs ongoing monitoring of which applications are active, so it can *dynamically change* the application of network connection handling (*stored WWAN differential traffic control policy*) based on a user’s interaction with applications running on the device (*monitoring of user interaction with the device*). See SAMSUNG-1005, [0057]; *see also* SAMSUNG-1016. In Shell, *user interactions with the device*, such as “activating a control in a user interface, for example, a button, a drop down list or other element...” are *monitored* in order to initiate requests to access a remote resource, where “the most appropriate connection method for connecting to the destination network (i.e. resource) may be identified” “automatically, “without requiring any user intervention at all” (*dynamically change the application of the stored WWAN differential traffic control policy*). SAMSUNG-1005, [0051]-[0052].

[21] The wireless end-user device of claim 1, wherein the one or more processors are configured to associate the particular application with the differential traffic control policy based on an application behavior.

132. As described in Section XI.A.1, a POSITA would have configured the Shell-Cole device to apply policies based on an application's present interaction state. For example, Shell explains that priorities may be "assigned to connection requests based on the current state of applications associated with the connection requests," e.g., "a request from an application that is currently active may be assigned a higher priority than a request from an application running in the background." SAMSUNG-1005, [0057]. A POSITA would have found obvious that a present interaction state of an application (foreground/background) represents an example of ***application behavior*** because an application's present interaction state indicates whether the application is currently made available for display on a mobile device.

133. This aligns with '359 patent's disclosure of "application behavior" of an application, such as "conference meeting services, video streaming, content update, software update," and "running in the background and continuing to consume potentially." SAMSUNG-1001, 8:23-31. Further, the Shell-Cole device would have been configured to apply connection request handling (i.e., the policy) to perform application-related prioritization based on the application's present

interaction state (*configured to associate the particular application with the differential traffic control policy based on an application behavior*).

[23.1] The wireless end-user device of claim 1, further comprising a network stack interface that intercepts network socket interface messages for applications and OS functions,

134. As discussed for [9], a POSITA would have understood and found obvious that the connection manager modules of Shell-Cole are part of the protocol stack (*network stack*) and configured to interact with several APIs, including system level APIs associated with different network connections and APIs used to communicate network information to application modules. SAMSUNG-1005, Abstract, [0048]; *see* SAMSUNG-1013, [0013], [0023], FIG. 2; *see also* SAMSUNG-1014, 45-56; SAMSUNG-1010, 16:23-30, 17:1-28.

135. A POSITA would have understood and found obvious that the connection manager modules of Shell Cole are configured to *intercept network socket interface messages because they are part of the protocol stack, which manages communication between applications and the operating system*.

SAMSUNG-1005, Abstract, [0048]; *see* SAMSUNG-1013, [0013], [0023], FIG. 2.

The protocol stack of Shell-Cole would have been configured to interact with several APIs, including system level APIs associated with different network connections and APIs used to communicate network information to application modules, or at least would have found such an implementation obvious based on

Shell's disclosure and the well-known nature of APIs and network stacks as common programming techniques that the '359 patent did not invent.

136. By interacting with system-level APIs associated with network connections, the modules handle network-related operations directly at the socket interface level. This configuration allows the connection manager to effectively monitor and manage data flow between applications and OS functions. The use of network stacks and APIs for interception of messages to ensure efficient network communication and control represented well-known techniques by the Critical Date.

[23.2] the one or more processors configured to apply the stored WWAN differential traffic control policy to at least some of the intercepted network socket interface messages.

137. See [23.1]. In Shell-Cole, processor 111 (*one or more processors*) would have been configured to perform processing associated with connection request handling functionality provided by the connection manager modules. SAMSUNG-1005, [0040]. As discussed for [1.4], the connection request handling functionality in Shell-Cole involves the application of network connection handling (*apply[ing] the differential traffic control policy*). SAMSUNG-1005, [0013], [0048], [0055], [0092].

138. As discussed for [23.1], this would have involved a protocol stack being configured to interact with several APIs, including system level APIs

associated with different network connections and APIs used to communicate network information to application modules, including effectively monitoring and managing data flow between applications and OS functions (*at least some of the intercepted network socket interface messages*). By intercepting network socket interface messages, the processors can analyze and enforce the differential traffic control policy, ensuring that certain network traffic is prioritized or managed according to predefined rules. This is confirmed by Rao, which describes similar types of prioritization techniques before the Critical Date. *See generally* SAMSUNG-1018. Such a configuration involves using well-known techniques in network optimization, which would have been a solution for controlling traffic in a WWAN environment.

[26] The wireless end-user device of claim 1, wherein the one or more processors are further configured to determine that the different application is an active application based on a user interaction with the different application.

139. Shell-Cole renders [26] obvious based on Shell's disclosure of monitoring application state in relation to network connection request handling. SAMSUNG-1005, [0057], [0092].

140. In Shell, connection manager modules may assign priorities to network connection requests "based on the current state of applications associated with the connection requests." SAMSUNG-1005, [0057]. For example, "a request

from an application that is currently active may be assigned a higher priority than a request from an application running in the background.” *Id.* Further, “voice requests may be assigned a higher priority than data requests.” *Id.* A POSITA would have therefore understood and found obvious that an application that provides a voice-related connection request (*different application*) involves a determination that a user is likely to interact with the application (e.g., voice call) (*a user interaction with the different application*), and thus, prioritization by the connection manager modules involves *determin[ing] that the different application is an active application*. For example, “if a user attempts to use a telephone line for a voice call, a user may need to cause a current data connection to disconnect before a voice call may be established.” SAMSUNG-1005, [0008]; *see id.*, [0013], [0014].

[27] The wireless end-user device of claim 1, further comprising an agent to block, modify, remove, or replace user interface messages generated by the particular application based on the applied WWAN differential traffic control policy.

141. As discussed in Section XI.A.1, Shell-Cole would have included connection manager modules (*agent*) for performing connection management handling. SAMSUNG-1005, [0057], [0092]. A POSITA would have configured the Shell-Cole device such that the connection manager modules prevent an application module from consuming excessive network resources. In Shell, this is

accomplished by providing notification data indicating that “a connection to a resource is blocked for some reason.” SAMSUNG-1005, [0092]. Conditions associated with blocking may include “bandwidth, latency, power consumption, cost, physical location, past connection success, or other parameters...” *Id.* A POSITA would have understood and found obvious that such blocking would have precluded an application module from sending user interface messages associated with the application’s connection request (***block, modify, remove, or replace...user interface messages***). For example, the restriction denies system resources and “CPU activity” that would have been necessary for an application module to generate and output user interface messages. Techniques for blocking applications from presenting user interface messages were also well-known by the Critical Date. *Id.* For example, Dive-Reclus (SAMSUNG-1010) teaches blocking an “unsigned” application and provides an interface over the blocked application asking whether it should be blocked. SAMSUNG-1010, 19:17-25, FIG. 2.

[28] The wireless end-user device of claim 1, wherein the one or more processors are configured to apply the stored WWAN differential traffic control policy to selectively block network access by the particular application by intercepting open, connect, and/or write requests by the particular application to a network stack.

142. Shell-Cole renders [28] obvious based on Shell’s disclosures regarding network connection handling, including blocking a connection request. SAMSUNG-1005, [0048], [0053], [0061], [0083], [0091], [0092], FIG. 2.

143. Shell describes that connection manager modules may provide notifications to application modules relating to changes in network connectivity, including a notification “that connection is being disconnected” or a notification “that a connection to a resource is blocked for some reason...” SAMSUNG-1005, [0092]. From this disclosure, a POSITA would have understood and found obvious that the connection manager modules are configured to provide network connection request handling (*apply the stored WWAN differential traffic control policy*) by blocking a network connection request provided by an application module (*selectively block network access by the particular application*). *Id.* For example, the network connection request may be blocked based on one or more “connection specific parameters,” such as “bandwidth, latency, power consumption, cost, physical location, past connection success...” *Id.*

144. As discussed above, because the connection manager modules of Shell-Cole are within a protocol stack between application modules and system level APIs, a POSITA would have understood and found obvious that blocking network connection requests involves intercepting network traffic of application modules (*intercepting open, connect, and/or write requests by the particular application to a network stack*). SAMSUNG-1005, [0048], [0053], [0061], [0083], [0091], FIG. 2.

145. Specifically, blocking a connection request by the connection manager modules of Shell-Cole would have involved “intercepting” the connection request because the act of blocking involves the detection and examination of the request before it reaches its intended destination or the network stack.

Additionally, interception allows the system to evaluate the request’s legitimacy and decide whether to allow or block it, thus controlling the flow of network traffic. By handling requests before they are transmitted over the network, the connection manager modules of Shell-Cole can accurately categorize the traffic, which becomes more challenging once the data is encrypted or mixed with other traffic. This early identification allows for precise traffic management and security enforcement, which is harder to achieve deeper in the network stack where data may be obscured. *See* SAMSUNG-1001, 60: 33-61:6.

B. GROUND 1B: Shell-Cole-Araujo Renders Claims 2, 16-18, 20, 27, and 29 Obvious

1. Combination

146. As discussed in Section XI.A.1, Shell-Cole provides functionality enabling the evaluation of network and power conditions using Cole’s techniques to determine whether to apply an Internet access control to connect to the network with the “best” connection type, then evaluates further conditions (e.g., whether an application is running in the foreground or background, which application is

running, etc.). SAMSUNG-1005, [0047], [0048], [0057], [0070], [0092];
SAMSUNG-1006, [0011], [0078], [0083], [0088], [0098].

147. In implementing such functionality, a POSITA would have perceived opportunities for improving the prioritization capabilities already present in Shell to impart additional advantages, such as increasing energy efficiency. To this aspect, a POSITA would have understood and found obvious that well-known monitoring capabilities may be useful to reduce power consumption and thereby increase the longevity of techniques provided by Shell-Cole while its device operates on battery power. *Id.* A POSITA would have thereby looked to Araujo’s “request deflector component,” which is “interposed” between each application and network driver, to selectively allow or block an application from transmitting data to the network. SAMSUNG-1008, [0050].

148. Incorporation of functionality of Araujo’s “request deflector component” into Shell-Cole would have enabled assignment of a “status” to an application, use of that status to determine if allowing an application request “justifies the consumption of power under the circumstances that are present,” and regulation (e.g., blocking, allowing) of the request based on that determination. SAMSUNG-1008, [0027]. Such a configuration would have thereby enjoyed the benefits of better using “energy in a way that strikes a balance between providing functionality and maintaining longevity of the charge.” *Id.*, [0001].

149. For example, in modifying Shell-Cole to incorporate Araujo's teachings, a POSITA would have "interposed" Araujo's "request deflector component" between Shell's application modules and device drivers. *See* SAMSUNG-1005, FIG. 2. For example, a POSITA would have arranged Araujo's "request deflector component" to process connection requests in relation to device power consumption. *Id.* This configuration would have enabled further assessment of whether the application can access the network (e.g., in addition to considering whether an application has exceeded application-specific data limitations) in order to further enhance the efficiency of the Shell-Cole-Araujo device.

150. A POSITA would have configured the "request deflector component" to determine whether each of the applications has a "status [that] justifies the consumption of power under the circumstances that are present," and selectively allow or block the network packets from proceeding to the network driver based on the determination. For example, upon determining that the application's status justifies the consumption of power, the "request deflector component" would allow the network packets to proceed to the network driver for prioritization. Alternatively, upon determining that the application's status does not justify the consumption of power, the "request deflector component" would block the network packets from reaching the network driver, either permanently or

temporarily. The “request deflector component” would have thereby provided an additional layer of control for regulating each of the applications’ access to the network (e.g., to supplement the regulation techniques already present in Shell and Cole). *Id.*

151. Finally, a POSITA would have had a reasonable expectation of success in modifying Shell-Cole to incorporate Araujo. The combination would involve using well-understood methods to integrate components with predictable results. *Id.* Specifically, combining Shell’s connection manager modules with Cole’s network evaluation techniques and Araujo’s request deflector component would follow known engineering practices, resulting in a cohesive system capable of efficiently managing network access and power consumption. *Id.* This approach aligns with established techniques for improving device efficiency and battery longevity, as Araujo’s component is designed to selectively control network access based on application status and power conditions. Such integration would have been anticipated by a POSITA to enhance similar devices by optimizing energy usage and network management in the same manner, thus providing a predictable and reliable improvement to the Shell-Cole system.

2. *Analysis*

[2] *The wireless end-user device of claim 1, wherein the API indicates the unavailability to the particular application of an Internet data service that is currently available by emulating network messaging.*

152. As discussed for [1.5], Shell-Cole would have provided connection manager modules that use one or more APIs as a “means” or “mechanism” to provide communications relating to prioritization. SAMSUNG-1005, [0092]; SAMSUNG-1006, [0110], [0166], [0190], [0205]. A POSITA would have found obvious that, in incorporating Araujo’s “request deflector component” into Shell-Cole (see Section XI.B.1), certain exemplary configurations involve the “request deflector component” being implemented to provide communications functionality as an API. Thus, the “request deflector component” renders the *API* obvious insofar as the “request deflector component” in Shell-Cole-Araujo would have provided the functionality recited in the claim (e.g., emulating network messaging) as being performed by the API, as further addressed below. *Id.*

153. APIs that were configured to emulate network messaging functionality between applications and drivers were well-known as of the Critical Date. SAMSUNG-1015, Abstract, FIG. 2, 3:465:10; SAMSUNG-1011, [0001], [0027], [0050]; SAMSUNG-1016, 1, 3. These APIs were designed to facilitate smooth and efficient communication between software applications and hardware drivers by replicating the necessary network protocols and message formats. This emulation allowed applications to interact seamlessly with hardware components

without needing to directly manage the complexities associated with hardware communication protocols.

154. The use of APIs to regulate network communications within a layered computing architecture was similarly well-known before the Critical Date.

SAMSUNG-1019, 11; *see also* SAMSUNG-1001, 91:25-67. This approach involved using APIs to manage and facilitate the interactions between different layers of a computing architecture, ensuring that network communications were handled in an organized and efficient manner. APIs played a crucial role in this context by providing standardized methods and protocols that allowed various software components to communicate with each other seamlessly across different layers of the architecture. This not only improved the modularity and scalability of the system but also enhanced the overall reliability and performance of network communications

155. As discussed in Section XI.B.1, in Shell-Cole-Araujo, “if the power state change is not allowable under [a] policy,” the device can “block the request” and “notif[y] [the program] that the request will not be processed.” SAMSUNG-1008, [0038], [0041]. Further, “[t]he program may also be notified that the reason for the request is that the device is off[,]” e.g., “the program may receive an error code or result code indicating that the request calls for a device that is not turned on.” *Id.*; *see id.*, [0040]. A POSITA would have understood and found obvious

that Shell-Cole-Araujo would have responded to an intercepted request by an application by notifying the application that the request will not be processed and/or transmitting an error code or result code to the application (*emulating network messaging*). In doing so, the combined device would have provided the application with a greater degree of information regarding its attempt to access a network, as taught by Araujo. *Id.*

156. Moreover, Araujo's use of an "error code or result code" is consistent with the '359 patent specification's disclosure of an "emulated network API" that blocks an application's connection attempt and sends "a message...back to the application" regarding the blocking, such as a message that "the application will understand and can interpret to indicate that the network access attempt was not allowed/blocked, that the network is not available, and/or to try again later for the requested network access." SAMSUNG-1001, 91:25-67.

[16] The wireless end-user device of claim 1, wherein the API instructs the device to transition to a power save state.

157. In Shell-Cole-Araujo, power management is based on monitoring, for example, power consumption (*power save state*). SAMSUNG-1008, Abstract, [0004]. Shell-Cole-Araujo would have enabled changes to the way applications operate based on monitoring a power state, which a POSITA would have understood and found obvious involves, in some instances, setting the device to

reduce power consumption (*transition to power save state*). SAMSUNG-1008, [0031], [0046]. For example, “if the system is, say, running on battery power, it may be possible to wait for [an] update until the system is plugged in so that energy can be saved to allow the user to work longer before running out of battery power.” SAMSUNG-1008, [0016]. Further, given that the connection manager modules of Shell-Cole-Araujo are configured to implement decision logic relating to network connection request handling using one or more APIs, it would have been obvious that such functionality is instructed by an API used for providing network connectivity information to an application (*the API*). SAMSUNG-1005, [0048]; SAMSUNG-1013, [0013], [0023], FIG. 21; SAMSUNG-1014, 45-461 SAMSUNG-1010, 16:23-30, 17:1-28.

[17] The wireless end-user device of claim 1, wherein the one or more processors are further configured to dynamically change the application of the stored WWAN differential traffic control policy based on a power state of the device.

158. In Shell-Cole-Araujo, power management is based on monitoring, for example, power consumption (*power state*). SAMSUNG-1008, Abstract, [0004]. Shell-Cole-Araujo would have enabled changes to the way applications operate based on monitoring a power state, which a POSITA would have understood and found obvious involves, in some instances, adjusting a policy enforcement by Shell-Cole-Araujo for prioritization (*dynamically change the application of the*

stored WWAN differential traffic control policy). SAMSUNG-1008, [0031], [0046]. For example, “[a] power state change may be sought in order to turn on a device, or to raise the power state of a device, in order to allow the device to service a request from an application.” SAMSUNG-1008, [0038]. In response, the Shell-Cole-Araujo device would respond by “block[ing] the request” and the “program may receive an error code or result code.” *Id.*, [0040]. One type of request is a network access request described in Shell, and a POSITA would have understood and found obvious that blockage of the request would have resulted in the Shell-Cole-Araujo device applying a different policy for prioritization (e.g., selecting a policy that prioritizes network access by another application that would not necessitate a power state change). SAMSUNG-1005, [0057], [0092].

159. Cole also describes selection of a network based on power consumption (e.g., “communication load monitoring”), and if connecting to a particular network “consumes significant battery power[,]” the connection manager 260 may “ask the user to find a better connection.” SAMSUNG-1006, [0088]. Cole further teaches that “[d]uring use, the mobile device 110 may be placed into a particular system configuration to emphasize particular performance characteristics,” including “battery life.” *Id.*, [0091]. Similarly, Araujo describes that “power manager 110 may cause devices to be turned off, or into a lower power state, based on factors such as timers, the amount of battery power remaining, or

any other factor.” SAMSUNG-1008, [0022]. From these disclosures, a POSITA would have understood and found obvious that the connection manager modules of Shell-Cole-Araujo would have similarly been configured to adapt connection request handling (*dynamically change the application of the stored WWAN differential traffic control policy*) in a manner that minimizes power consumption by placing a mobile device into a state that conserves power (*power state*), e.g., a state where devices are turned off and/or the device is permitted to only connect to certain types of networks.

[18] The wireless end-user device of claim 1, wherein the one or more processors are further configured to dynamically change the application of the stored WWAN differential traffic control policy based on a device usage state.

160. Shell-Cole-Araujo incorporates the disclosures of Shell and Cole and thereby render this claim obvious for the reasons discussed above in Ground 1A. See SAMSUNG-1005, [0033], [0051], [0052], [0057]. Araujo further describes policy application based on “power usage” (*device usage state*). SAMSUNG-1008, Abstract, [0002], [0019], [0032]. For example, Araujo describes predicting events relating to “a known pattern of usage,” e.g., “a particular daemon with high (e.g., ‘VIP’) status known to use the network card every 30 seconds,” which is then accounted for in “determining whether to change a given device’s power state.” *Id.*, [0033]. Thus, in at least this example, a change to network connection

handling by Shell-Cole-Araujo (*dynamically change the application of the stored WWAN differential traffic control policy*) is based on a known pattern of usage associated with a power state (*device usage state*).

[20] The wireless end-user device of claim 1, wherein the one or more processors are further configured to dynamically change the application of the stored WWAN differential traffic control policy based on power control state changes for one or more of the modems.

161. As discussed for [17], a POSITA would have understood and found obvious that Shell-Cole-Araujo adjusts policy enforcement for network access in certain circumstances involving a power state change. Araujo further describes that “power may be controlled for devices,” such as “disks, network adapters, processors, monitors.” SAMSUNG-1008, [0018]. A POSITA would have understood and found obvious that a modem is an example of network adapter, and thus, Shell-Cole-Araujo would have provided power management for *one or more modems*. A POSITA would have therefore understood and found obvious that, in some instances, Shell-Cole-Araujo adjusts policy enforcement for network access based on a power state change for a modem (*dynamically change the application of the stored WWAN differential traffic control policy*). SAMSUNG-1008, [0031], [0046]; SAMSUNG-1006, [0088]-[0091]. This involves dynamically altering the application of the first differential traffic control policy to prioritize tasks according to the modem's current power state. For example, if the modem

transitions from a high power state to a low power state, the system might modify the traffic control policy to prioritize essential communications and reduce non-critical data transfers, thereby conserving power while maintaining necessary network functionality. This dynamic adjustment is crucial for optimizing both performance and energy efficiency, particularly in mobile and portable devices where power management is a significant concern. The POSITA would recognize that leveraging the power state of the modem to influence traffic control policies aligns with established principles of resource management and power conservation. By implementing such adaptive policies, the combined system ensures that network resources are used efficiently, extending the operational life of the device while still maintaining connectivity and performance standards.

[27] The wireless end-user device of claim 1, further comprising an agent to block, modify, remove, or replace user interface messages generated by the particular application based on the applied WWAN differential traffic control policy.

162. Shell-Cole-Araujo incorporates the disclosures of Shell and Cole and thereby render this claim obvious for the reasons discussed above in Ground 1A.

See SAMSUNG-1005, [0057], [0092]; SAMSUNG-1010, 19:17-25, FIG. 2.

Araujo further describes a “request deflector component” (***agent***) that provides various power management features. *See* SAMSUNG-1008, [0024]

(“intercept[ing] requests and respond[ing] to the requests based on power

management considerations before the requests reach a device driver”); *see id.*, [0001], [0050]. For example, the request deflector component may deflect (***block, modify, remove, or replace***) certain requests generated by application modules (including ***the particular application***) based on network connection request handling (***applied WWAN differential traffic control policy***).

163. A POSITA would have also found it obvious requests provided by the application modules may relate to user interactions (***user interface messages generated by the first end-user application***) such that the request deflector component may deflect requests related to such user interactions. For example, Shell describes that a “request to access a remote resource” may be provided based on a user “[a]ctivating a control in a user interface,” e.g., “a button, a drop down list or other element.” SAMSUNG-1005, [0051]. In this example, a resulting request that is deflected by the request deflector component involves ***block[ing]...user interface messages generated by the first end-user application.***

Further, as discussed in Ground 1A, techniques for blocking applications from sending user interface messages were well-known by the Critical Date. For example, Dive-Reclus (SAMSUNG-1010) teaches blocking an “unsigned” application and provides an interface over the blocked application asking whether it should be blocked. SAMSUNG-1010, 19:17-25, FIG. 2.

[29] The wireless end-user device of claim 2, wherein emulating network messaging comprises responding to a network request from the particular application by blocking the request from passing to a network stack and returning to the particular application a message indicating the network request was not successful.

164. As discussed for [21], a POSITA would have configured Shell-Cole-Araujo to respond to an intercepted request from an application by notifying the application that the request will not be processed and/or transmitting an error code or result code to the application (*emulating network messaging*). Further, as discussed in Section XI.B.1, Shell-Cole-Araujo would have incorporated Araujo’s “request deflector component” to “intercept[] requests and respond[] to the requests based on power management considerations before the requests reach a device driver.” SAMSUNG-1008, [0024], [0050].

165. A POSITA would have also understood and found obvious to configure Shell-Cole-Araujo such that, in blocking a request from reaching the network driver, the request is prevented from being provided to Shell’s connection manager modules (which are included in the network stack) (*blocking the request from passing to a network stack*). SAMSUNG-1005, [0057], [0092].

166. Further, in Shell-Cole-Araujo, “if the power state change is not allowable under the policy,” the device can “block the request” and “notif[y] [the program] that the request will not be processed.” SAMSUNG-1008, [0038], [0041]. Further, “[t]he program may also be notified that the reason for the request

is that the device is off[,]” e.g., “the program may receive an error code or result code indicating that the request calls for a device that is not turned on” (...*and returning to the particular application a message indicating the network request was not successful*). *Id.*, [0040].

C. GROUND 1C: Shell-Cole-Freund Renders Claims 3-5 Obvious

1. Combination

167. Shell and Freund describe regulating access to network traffic. As discussed above, in Shell, connection manager modules “may utilize a fairly complex heuristic to attempt to prioritize among multiple competing connection requests while balancing priority with performance and fairness among requests.” SAMSUNG-1005, [0018]. Such prioritization may involve “choos[ing] to preempt a higher priority request if by doing so it can honor a set of lower priority requests, or alternatively if the higher priority request has exceeded some time or bandwidth allotment, or for some other reason.” *Id.* Further, the connection manager modules may “analyze[] a variety of factors to determine the most appropriate connection method to connect to the destination network,” including “the local resources currently available on the computing device, the priority of the request, or connection parameters (such as latency, cost, and bandwidth) associated with a connection method.” *Id.*, [0033]. As another example, the connection

manager modules may consider various other parameters, such as “past connection successes.” *Id.*, [0016], [0092].

168. A POSITA would have recognized that the network traffic control policies of Freund are simply additional examples of prioritization techniques contemplated by Shell. For example, Freund includes “access rules [that] can include criteria such as total time a user can be connected to the Internet ... time a user can interactively use the Internet ... a list of applications or application versions that a user can or cannot use in order to access the Internet, a list of URLs (or WAN addresses) that a user application can (or cannot) access, a list of protocols or protocol components ... that a user application can or cannot use, and rules to determine what events should be logged.” SAMSUNG-1012, 4:9-19. A POSITA would have naturally recognized that Freund’s policy examples are among the techniques for “comparing prioritized connection requests to local resources available in a local computing device,” as described in Shell. SAMSUNG-1005, Abstract.

169. Shell also describes embodiments in which “service technicians may be able to configure connection methods in a manner that is suitable to entities that have an interest in connection methods.” SAMSUNG-1005, [0049]. This may be accomplished using “configuration user interface 201” to “configure connection methods.” *Id.* For details on the configuration interface 201, a POSITA would

have looked to references, such as Freund, which shows “bitmap screenshots illustrating a preferred user interface or ‘wizard’ dialog for configuring rules” including a preferred interface that “provides a ‘view’ of rules governing operation of the Internet access monitor, displaying all of the rules which are available for a current configuration.” SAMSUNG-1012, 6:47-49, 24:17-20, FIGS. 7A-K. Such a UI allows a user to create, edit, or delete rules. SAMSUNG-1012, 24:22-24.

170. Shell also explains “decision logic utilized to allocate accessible resources to connection requests.” SAMSUNG-1005, [0058]. For implementation details on such decision logic, a POSITA would have looked to references such as Freund. As described above, Freund teaches an example of policies being loaded by an agent: “the Supervisor filters rules appropriate for the client (i.e., application, Host, and other rules), and transmits them to the Client Monitor.” SAMSUNG-1012, 28:14-18.

171. Because both Shell and Freund teach using policies/rules to control network traffic, including defining and loading policies/rules, it would have been obvious to a POSITA to combine the teachings of Shell and Cole with Freund’s teachings to achieve the benefits of Freund’s UI for configuring rules and for more detailed enumeration of network traffic control policies contemplated but not explicitly described in Shell.

172. In operation, the resulting Shell-Cole-Freund device would have used the policy-based network selection techniques of Cole to determine the best network connection and use the traffic management techniques of Shell and Freund to use that network efficiently. SAMSUNG-1006, [0062]; SAMSUNG-1005, [0057], [0092]; SAMSUNG-1009, 4:9-19. In addition, the policies governing the operation of the Shell-Cole-Freund device would be configured with the UI techniques of Freund and/or loaded from a network element using the techniques of Shell and Freund. SAMSUNG-1005, [0062]; SAMSUNG-1012, 24:22-24. The components of Shell-Cole-Freund each operate as they would in isolation.

173. A POSITA would have a reasonable expectation of success in creating the combination, which would have required only routine programming knowledge well within the skill of a POSITA before the Critical Date. *Id.* The change would have amounted to nothing more than the use of a known technique to improve similar mobile devices in a similar way and combining prior art elements according to known methods to yield the predictable results described above. *Id.* The processor of the Shell-Cole-Freund device would have been configured to perform the operations discussed in this section.

2. *Analysis*

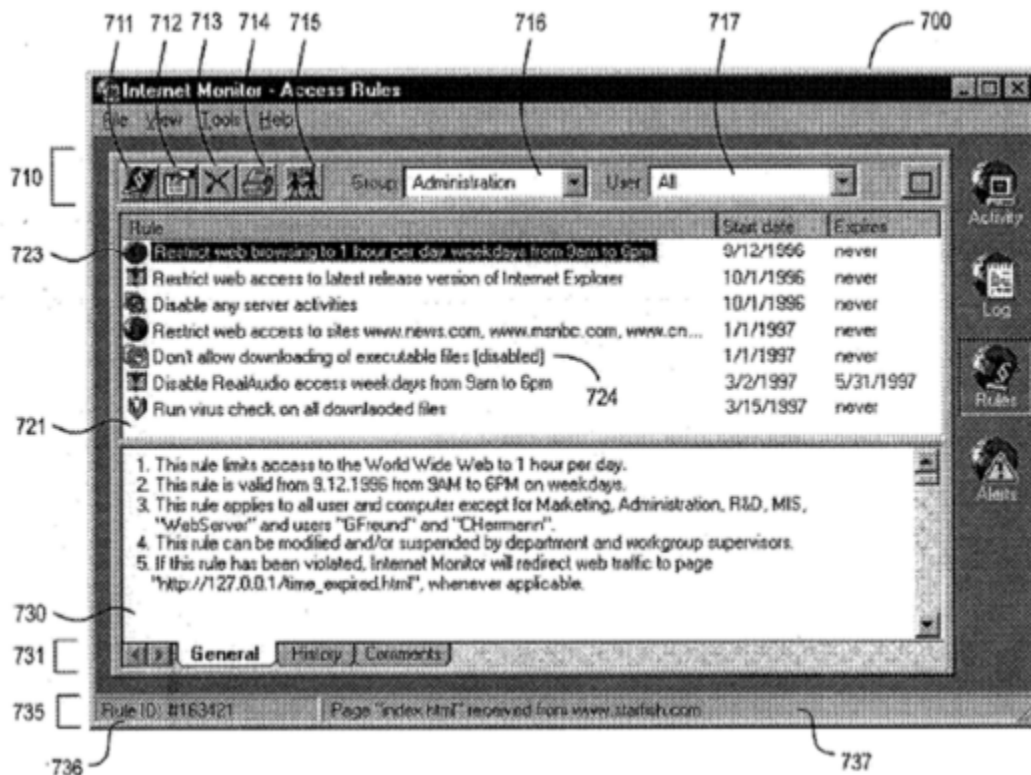
[3] The wireless end-user device of claim 1, wherein the one or more processors are further configured to apply the differential traffic control policy to the particular application when a user of the device is not directly interacting

with the particular application or perceiving any benefit from the particular application.

174. Shell-Cole-Freund renders [3] obvious based on Freund's disclosure of user-device interactions. SAMSUNG-1012, 3:15-16, 10:14-24. In Freund, "'background' use" is defined as "when an application accesses the Internet without user interaction." SAMSUNG-1012, 3:15-16. Freund additionally contrasts active and background use, explaining that active use "occurs when a user directly interacts with an Internet application (e.g., Web browser) while that application accesses the Internet" while background use "occurs when an application or process executing in the background (i.e., does not have 'focus') accesses the Internet, such as a mail client which intermittently polls an Internet-based mail server." *Id.*, 10:14-24. Since Freund defines background use as occurring when an application uses background data services, i.e., Internet access without user interaction, then by Freund's definition, an application is classified as not interacting with a user (***when a user of the device is not directly interacting with the particular application***) when the application module uses background data services. A POSITA would have found it obvious that Shell's application of the differential traffic control policy to the particular application when operating in the background includes and/or applies to Freund's description of background applications as those lacking direct user interaction. *Id.*

[4] The wireless end-user device of claim 1, further comprising a user interface to provide a user of the device with information regarding why the differential traffic control policy is applied to the particular application.

175. Shell-Cole-Freund renders [4] obvious based on Freund's disclosure of user interfaces to configure policy application. SAMSUNG-1005, [0014], [0038]-[0041], [0044], [0049]; SAMSUNG-1012, 24:16-20, 25:1-13, FIGS. 7A-K. For example, Freund's Figure 7A shows an example of a user interface:



SAMSUNG-1012, FIG. 7A

Freund explains, "FIGS. 7A-K illustrate a preferred user interface or 'wizard' dialogs for configuring rules," and in FIG. 7A, a preferred interface "provides a 'view' of rules governing operation of the Internet access monitor, displaying all of

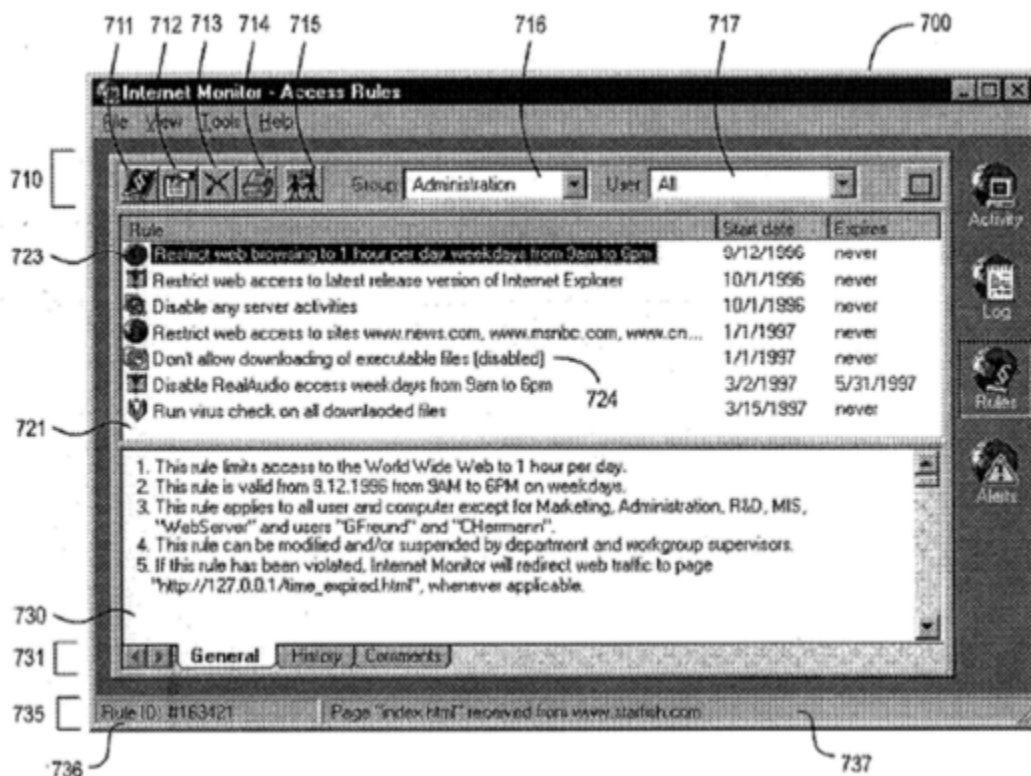
the rules which are available for a current configuration.” SAMSUNG-1012, 24:16-20. The policies (rules) can be associated with applications using the “wizard dialog 740 [that] asks the user what kind or type of new rule should be created.” *Id.*, 25:1-3, FIGS. 7A-K. Freund explains that “any given rule is a combination of access rights granted on the basis of available applications, permitted time limits, permitted user activities, permitted protocols, and the like.” *Id.*, 25:3-6. “[T]he user can select type 742 for limiting what applications (including individual applications) can do on the Internet.” *Id.*, 25:10-13.

176. Further, interface 700 includes a “rule details panel 730” that “provides detail information for [a] particular rule which is currently selected in the panel 721,” e.g., “general features of the rule,” “on-line activity,” and a “status message indicating which objects are currently being received from a Website.” *Id.*, 24:45-63. From these disclosures, a POSITA would have understood and found obvious that the various types of information presented on the Freund interface (e.g., access rights as Internet activity access controls) (rule information, activity information associated with a rule) are examples of ***information regarding why the first differential traffic control policy is applied to the first end-user application***).

[5] The wireless end-user device of claim 1, further comprising a user interface to inform a user of the device when there are options to set,

control, override, or modify service usage controls that affect the differential traffic control policy.

177. Shell-Cole-Freund renders [5] obvious based on Freund's disclosure of user interfaces to configure policy application. SAMSUNG-1005, [0014], [0038]-[0041], [0044], [0049]; SAMSUNG-1012, 24:16-20, 25:1-13, FIGS. 7A-K. For example, Freund's Figure 7A shows an example of a user interface:



SAMSUNG-1012, FIG. 7A

Freund explains, "FIGS. 7A-K illustrate a preferred user interface or 'wizard' dialogs for configuring rules," and in FIG. 7A, a preferred interface "provides a 'view' of rules governing operation of the Internet access monitor, displaying all of the rules which are available for a current configuration." SAMSUNG-1012,

24:16-20. The policies (rules) can be associated with applications using the “wizard dialog 740 [that] asks the user what kind or type of new rule should be created.” *Id.*, 25:1-3, FIGS. 7A-K. Freund explains that “any given rule is a combination of access rights granted on the basis of available applications, permitted time limits, permitted user activities, permitted protocols, and the like.” *Id.*, 25:3-6. “[T]he user can select type 742 for limiting what applications (including individual applications) can do on the Internet.” *Id.*, 25:10-13. A POSITA would have recognized this functionality informs the user of and allows a user to adjust various settings (options to set, control, override, or modify service usage controls) relating to access rights (differential traffic control polic[ies]).

D. GROUND 1D: Shell-Cole-Montemurro Renders Claims 6-8, 12, 15, 22, 24, And 25 Obvious

1. Combination

178. As described in Section XI.A.1, a POSITA would have configured Shell-Cole to apply policies to prioritize packet transmission and application functionality to achieve several types of operational efficiencies. SAMSUNG-1005, [0007], [0057], [0092]; SAMSUNG-1006, [0011]. In implementing this combination, a POSITA would have recognized opportunities to provide additional efficiencies in enabling access to multiple networks, as contemplated by each of Shell and Cole (and thereby provided by Shell-Cole). *Id.* This would have

motivated a POSITA to seek solutions beyond Shell and Cole to achieve such additional efficiencies. *Id.*

179. Given Montemurro's focus to "optimize communications using a policy based mechanism to configure connections and routes," a POSITA would have found Montemurro's teachings to be applicable to the policies applied by Shell-Cole for network access regulation. Like Shell, Montemurro describes embodiments where its policies are used to "disable or enable an application for a user." SAMSUNG-1009, [0027]. This would have led a POSITA to consider Montemurro's teachings in Figure 2 relating to "policy-based data routing for multimode operations." *Id.*, [0024]. Incorporation of such teachings into Shell-Cole would have yielded several improvements described in Montemurro, such as the ability to "offer different usage models depending on the mode of wireless operation selected." *Id.*, [0003]. This configuration would have improved Shell-Cole by enabling prioritization to further account for several considerations, such as "bandwidth, range, cost, and power consumption." *Id.* Further, policies applied by Shell-Cole-Montemurro would have been "responsive to various factors such as Radio Access Technology (high/low bandwidth), cost, presence, time of day, location, application type and quality of service (QoS) requirements among others to optimize communications," as described in Montemurro. *Id.*, [0011].

180. Finally, a POSITA would have had a reasonable expectation of success in modifying Shell-Cole to incorporate Montemurro. The combination would have involved using well-understood methods to integrate components with predictable results. Specifically, combining Shell's connection manager modules with Cole's network evaluation techniques and Montemurro's network selection techniques would follow known engineering practices, resulting in a cohesive system capable of efficiently managing network access and power consumption for different network types. *Id.* This approach aligns with established techniques for improving network connection request handling for different network types, as Montemurro's policy-based routing techniques enable access to multiple network types. *Id.* Such integration would have been anticipated by a POSITA to enhance similar devices by optimizing energy usage and network management in the same manner, thus providing a predictable and reliable improvement to the Shell-Cole system. *Id.*

2. *Analysis*

[6] *The wireless end-user device of claim 1, wherein the differential traffic control policy is part of a multimode profile having different policies for different networks.*

181. Shell-Cole-Montemurro would have applied a rules engine to coordinate connectivity for applications. SAMSUNG-1009, [0027]. As described in Section XI.D.1, in certain scenarios, a rules engine would have applied an

instance of a routing table (*multimode profile*) with information for multiple networks available for selection. *Id.*, [0030]. For example, “if both the WLAN and WWAN radios...of [a] device [] are connected to their respective networks 104 and 106, there will be a route associated with each network 104 and 106...” *Id.* The rules engine “executes” the instance of the routing table “to ensure that data goes out to the most appropriate network” and also to “determine which interface would be best used to service a particular application.” *Id.*, [0035].

182. Further, in scenarios where multiple networks are available for selection, POSITA would have understood and found obvious that the information specified by the instance of the routing table includes policies for each available network. In Montemurro, policies are “responsive to various factors such as Radio Access Technology, high/low bandwidth, cost, presence, time of day, location, application type and quality of service (QoS) requirements.” SAMSUNG-1009, [0011], [0024], [0027], [0029]. Thus, a POSITA would have understood and found obvious that an instance of the routing table with information for multiple networks available for selection includes “different policies for different networks.” SAMSUNG-1009, [0011], [0024], [0027], [0029].

183. The use of different policies for enabling capabilities for different networks (e.g., phone network, local network) was conventional by the Critical Date. *See* SAMSUNG-1010, 12:13-13:9). For example, Dive-Reclus

(SAMSUNG-1010) includes a table showing different permission decisions for applications submitting API calls to request network access. By implementing distinct policies, network administrators could enhance performance and security, ensuring that each network type operated optimally within its intended context. Furthermore, this differentiation facilitated better resource management and improved user experiences across various network environments.

Installed	PhoneNetwork	ReadUserData	WriteUserData	LocalNetwork	Certificate
Good.exe	Not granted	Granted	Not granted	Granted	Checked OK – click to view root signing cert
Bad.exe	Not granted	Not granted	Not granted	Not granted	Unsigned

SAMSUNG-1010, 20

[7] The wireless end-user device of claim 6, wherein the one or more processors are further configured to select a traffic control policy from the multimode profile based at least in part on the type of network connection currently in use by the device.

184. See [6]. A POSITA would have also understood and found obvious that, in some scenarios, information specified by an instance of the routing table (*multimode profile*) includes “different policies for different networks.”

SAMSUNG-1009, [0011], [0024], [0027], [0029]. The rules engine “executes” the instance of the routing table “to ensure that data goes out to the most appropriate

network” and also to “determine which interface would be best used to service a particular application.” *Id.*, [0035]. For example, “if both the WLAN and WWAN radios...of [a] device [] are connected to their respective networks 104 and 106, there will be a route associated with each network 104 and 106...” *Id.*, [0030]. Thus, in this example, a POSITA would have understood and found obvious that the rules engine within Shell-Cole-Montemurro would have executed the instance of the routing table based on types of current network connections of a device (WLAN, WWAN), and thus, when a routing option is selected from the routing table instance, the selection (*select a traffic control policy*) is based on the types of current network connections of the device (*type of network connection currently in use by the device*). Dive-Reclus (SAMSUNG-1010) also demonstrates that permission decisions regulating network access by third-party applications that use API calls were well-known by the Critical Date. *See* SAMSUNG-1010, 16:23-17:28, 20.

[8] *The wireless end-user device of claim 7, wherein the one or more processors are further configured to, when the type of network connection is at least one type of WLAN connection, select a traffic control policy from the multimode profile based at least in part on a type of network connection from the WLAN to the Internet.*

185. Montemurro offers an example in which, “if both the WLAN and WWAN radios...of [a] device [] are connected to their respective networks 104 and 106, there will be a route associated with each network 104 and 106...”

SAMSUNG-1009, [0030]. Thus, in this example, a POSITA would have understood and found obvious that the rules engine within Shell-Cole-Montemurro would have executed the instance of the routing table (*select a traffic control policy*) when the device is connected to a WLAN connection (*when the type of network connection is at least one type of WLAN connection*).

186. Further, Montemurro references a “local LAN” and a POSITA would have understood and found obvious that the “local LAN” is a type of *network connection from the WLAN to the Internet*. SAMSUNG-1009, [0039]. A POSITA would have also understood and found obvious that Shell-Cole-Montemurro enables connectivity to several types of network connections implemented by network components in the combination (e.g., base station, WLAN components, access points, etc.) between the WLAN and Internet such that the device considers a *type of WLAN connection* in *selecting a traffic control policy*, as discussed above. See SAMSUNG-1009, [0013].

187. The configuration discussed above represents an obvious configuration as this reduces costs and complexity, consistent with Montemurro’s teachings. This configuration allows for more efficient management of network resources by automatically selecting the most suitable connection based on factors such as cost, signal strength, and user preferences. Additionally, by implementing such a policy, network providers can optimize service delivery and enhance user

experience, particularly in scenarios where connectivity may fluctuate or where multiple network options are available.

[12] The wireless end-user device of claim 1, wherein the one or more network access conditions indicated via the API to the particular application comprises information on whether a current connected WWAN is a roaming network or a non-roaming network.

188. See [25.1], [25.2], *infra*. Montemurro identifies several types of WWAN, including “cellular technologies.” SAMSUNG-1009, [0003]. Policies are “responsive to various factors” (*id.*, Abstract) including “network roaming” (*id.*, [0029]). Further, the rules engine “executes” the routing table to “ensure that data goes out to the most appropriate network (via respective network interface)” and “interacts with the connection API’s to determine which interface would be best used to service a particular application.” *Id.*, [0035]. Given such non-limiting disclosure and the types of examples identified, a POSITA would have understood and found obvious that an instance of the routing table includes “information on whether a connected WWAN is a roaming network or a non-roaming network.” SAMSUNG-1009, [0022], [0029], [0037].

[15] The wireless end-user device of claim 1, wherein the API instructs the particular application to transition to a different state.

189. Shell-Cole-Montemurro would have incorporated an improved agent that would have leveraged several types of well-known APIs (including the recited *API*) in regulating applications. SAMSUNG-1005, [0048]; SAMSUNG-1011, 3-4;

see [1.5]. Dive-Reclus, for example, describes a well-known implementation of using permission decisions in regulating network access by third-party applications that use API calls. SAMSUNG-1010, 16:23-17:28, 20. A POSITA would have found it obvious that a permission decision denying network access results in *transition to a different state* (e.g., permitted network access, restricted network access), which is communicated to the application through a corresponding API (*API instructs the first end-user application*). SAMSUNG-1010, 16:23-17:28, 20.

190. Further, Montemurro provides techniques that cause an application to transition to a different state, e.g., “[s]ome access technology switches will result in IP address changes” so “an application using the connection needs to be aware of the IP address, the change may not be transparent and may need to be communicated to (or otherwise discoverable by) the application.” SAMSUNG-1009, [0048]. In another example, connection “differences are significant (e.g. between GSM/GPRS and WAN 802.11 technologies)” so “an application may need to tailor its behaviour to account for the new access technology in use” such as “[b]uffer sizes, re-transmission timers etc.” that “may need to be changed.” *Id.* A POSITA would have understood and found obvious scenarios where an application tailors its behavior (e.g., IP address changes, altered buffer sizes, and modified re-transmission timers), which serve as examples of application transitions to different states.

191. Thus, when an application state changes such that a network access condition adjusts the application's permission to access network resources, a POSITA would have understood and found obvious that, in Shell-Cole-Montemurro, an API instructs the application to transition states relating to the network access condition (*API instructs the first end-user particular application to transition to a different state*). SAMSUNG-1011, 3-4; see SAMSUNG-1001, 73:47-54.

192. For instance, if an application initially has full access to network resources but the network access condition changes due to a shift in user activity, network policy updates, or power management considerations, the API will instruct the application to adjust its state. This might involve reducing its network usage, altering its data transmission rates, or even suspending network operations altogether to comply with the new conditions. This dynamic state management is crucial for maintaining optimal network performance and resource allocation. The API serves as a critical intermediary, translating the network access conditions into actionable instructions for the application, thereby enabling real-time adjustments that enhance both efficiency and security.

[22] The wireless end-user device of claim 1, wherein the differential traffic control policy defines that applications to which the policy applies can only have WWAN network access events during particular time windows.

193. In Shell-Cole-Montemurro, policies are “responsive to various factors,” (SAMSUNG-1009, Abstract), including “cost, presence, time of day, location, e.g., geo-based policies, network roaming” (*id.*, [0029]). Further, the rules engine “executes” the instance of the routing table to “ensure that data goes out to the most appropriate network (via respective network interface)” and “interacts with the connection API’s to determine which interface would be best used to service a particular application.” *Id.*, [0035]. Thus, a POSITA would have understood and found obvious that Shell-Cole-Montemurro regulates access to connection events to several networks, including a WWAN network (*WWAN network access events*).

194. Further, as discussed above for [1.5] and Section XI.A.1, Shell-Cole-Montemurro would have regulated network access by applications through use of policies. In Montemurro, “[p]olicies may be responsive to various factors,” including “time of day.” SAMSUNG-1009, Abstract. A POSITA would have understood and found obvious that a policy applied by Shell-Cole-Montemurro regulates both the type of network access and the circumstances during which an application can have access to network services such that Shell-Cole-Montemurro “can only have WWAN network access during particular time windows” (e.g.,

time windows during which a policy specifies network connectivity). SAMSUNG-1009, [0035]. Moreover, use of policies to limit network access during particular time windows were conventional by the Critical Date. *See* SAMSUNG-1012, 10:13-17, 30:59-67.

195. For example, time-based access control was discussed by Joshi (SAMSUNG-1028) as a general policy mechanism. Specifically, “[i]n many organizations, processes and functions may have limited time spans or have periodic temporal durations. For instance, a part-time staff member in an organization may be authorized to work only on working days between 9:00 a.m. and 1:00 p.m.” SAMSUNG-1028, 1. A POSITA would find it obvious to extend the general time-based access control infrastructure introduced by Joshi to network access.

196. These policies were often implemented to manage network congestion, ensuring that resources were allocated efficiently and that the user experience remained optimal during peak usage times. For instance, restricting access to certain applications or services during high-traffic hours helped to maintain overall network performance and reduce latency for critical tasks. Such time-based access controls could also enhance security by minimizing exposure to potential threats during vulnerable periods. This approach demonstrated a

proactive strategy in network management, allowing administrators to balance user demands with system capabilities effectively.

[24] The wireless end-user device of claim 1, wherein the one or more processors are further configured to update the stored WWAN differential traffic control policy based on information received from a network element.

197. As discussed in Section XI.D.1, policies applied by Shell-Cole-Montemurro (including *the first differential traffic control policy*) would have been “responsive to various factors such as Radio Access Technology (high/low bandwidth), cost, presence, time of day, location, application type and quality of service (QoS) requirements among others to optimize communications,” as described in Montemurro. SAMSUNG-1009, [0011].

198. In Montemurro, “network infrastructure” provides support for services provided to a wireless device. SAMSUNG-1009, [0014]. Examples of “network infrastructure” include “a gateway server, a provisioning server, a discovery server, and an application repository,” each of which are examples of a “network element.” *Id.* Services provided by the network infrastructure include “Administrative and Management Service dealing with policies, such as those specifying allowed applications for users, services available to applications and more.” SAMSUNG-1009, [0015]-[0016].

199. Additionally, in Shell, “a variety of factors” are analyzed to “determine the most appropriate connection method to connect to the destination network,” which would have adjusted network connection handling based on, for example, “the local resources currently available on the computing device, the priority of the request, or connection parameters (such as latency, cost, and bandwidth) associated with a connection method.” SAMSUNG-1005, [0033]; *see id.*, [0092]. A POSITA would have understood and found obvious, where a server (*network element*) provides an update for a service dealing with policies, Shell-Cole-Montemurro updates policies relating to application operation (*update the first differential traffic control policy*) based on the update from the server (*information received from [the] network element*).

200. For example, the update may specify a new allowed application, and Shell-Cole-Montemurro would have updated policies to reference a name of the new allowed application and thereby permit regulation of the newly allowed application. SAMSUNG-1009, [0014]. Moreover, techniques for adapting policy configurations based on information received from network elements were conventional by the Critical Date. SAMSUNG-1012, Abstract, 5:9-12, 28:6-16. These techniques involved dynamically adjusting network policies in response to real-time data and feedback from various network components, such as routers, switches, access points, and other devices that comprise the network infrastructure.

201. By the Critical Date, it was common practice for network management systems to collect and analyze information from network elements to make informed decisions about policy adjustments. This information could include data on network traffic patterns, bandwidth usage, device performance, security events, and other relevant metrics. For example, if a network element detected a sudden spike in traffic, the management system could adapt the quality of service (QoS) policies to prioritize critical applications and ensure optimal performance.

[25.1] The wireless end-user device of claim 1, the WWAN differential traffic control policy comprising first and second differential traffic control policies applicable respectively to Internet data service provided using the WWAN modem to connect to a home WWAN and a roaming WWAN,

202. In Shell-Cole-Montemurro, “if both the WLAN and WWAN radios...are connected to their respective networks 104 and 106, there will be a route associated with each network 104 and 106” (below). SAMSUNG-1009, [0030]; *id.*, [0011], [0024], [0027], [0029].

Source IP	Destination IP	Device Interface (208)
192.168.1.20	0.0.0.0	WLAN (208A)
67.69.20.142	0.0.0.0	WWAN (208C)

203. In Montemurro, a device “has mobile IP capabilities” to “permit the device to attach to the Internet (IP network) via a home and one or more foreign

networks.” SAMSUNG-1009, [0037]. Since Montemurro contemplates devices with capabilities to connect to both home and foreign networks, a POSITA would have found obvious that the routing table may be similarly configured to provide routing options for various types of networks, including a *home WWAN* and a *roaming WWAN*. SAMSUNG-1007, [0037]. Further, as discussed for [25.2], *infra*, connection factors associated with each type of network represent examples of *first and second differential traffic control policies*. SAMSUNG-1009, [0011].

204. Montemurro’s device is operable on various networks (including WWAN), so a POSITA would have understood and found it obvious that the routing table is similarly configured to provide routing options for WWAN for a home network (*home WWAN*) and WWAN for roaming to a foreign network (*roaming WWAN*). SAMSUNG-1009, [0030], [0037].

205. By the Critical Date, WWAN was well-known for being implemented for different settings, such as a home WWAN and a roaming WWAN. Montemurro further offers non-limiting disclosure of WWAN, e.g., “cellular technologies,” which a POSITA would have understood and found obvious includes roaming networks and home networks. SAMSUNG-1009, [0029]-[0038]. A POSITA would have perceived network selection in Shell-Cole-Montemurro to provide connectivity to one of several available WWAN networks (*connect to a home WWAN and a roaming WWAN*) to represent an obvious configuration as

this reduces costs and complexity, consistent with Montemurro. For example, it was well-known and obvious to use and make selective connection to a home network (e.g., *home WWAN*) and a roaming network (e.g., *roaming WWAN*). SAMSUNG-1017, [0003].

[25.2] wherein the one or more processors are further configured to apply the first differential traffic control policy when Internet data service is provided through a home WWAN and to apply the second differential traffic control policy when Internet data service is provided through a roaming WWAN.

206. In Montemurro, each policy specifies a set of connection factors, such as “Radio Access Technology, high/low bandwidth, cost, presence, time of day, location, application type and quality of service (QoS) requirements.” SAMSUNG-1009, [0011]. A POSITA would have understood and found obvious that a set of connection factors specified within a policy represents a ***first differential traffic control policy*** in that they permit connectivity to an associated network, and thereby enable Montemurro’s “policy-based mechanism to configure connections and routes.” *Id.*, Abstract. Thus, in an instance of the routing table with routing options for a home network and a foreign network, the routing table includes a first set of connection factors for the home network (***first differential traffic control policy***) and a second set of connection factors for the foreign network (***second differential traffic control policy***). SAMSUNG-1009, [0011], [0024], [0027], [0029]. Thus, hierarchical policy structures for adjusting the types of device

functionality provided by Shell-Cole-Montemurro were conventional by the Critical Date.

XII. CONCLUSION

207. The findings and opinions set forth in this declaration are based on my work and examinations to date.

208. I may continue my examinations. I may also receive additional documentation and other factual evidence over the course of this IPR that will allow me to supplement and/or refine my opinions. I reserve the right to add to, alter, or delete my opinions and my declaration upon discovery of any additional information. I reserve the right to make such changes as may be deemed necessary.

209. In signing this declaration, I recognize that the declaration will be filed as evidence in an IPR before the PTAB. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

210. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the

knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Declaration of Kevin R. B. Butler
SAMSUNG-1003

Dated: Sep 6, 2024

By: 
Kevin R. B. Butler

APPENDIX A

KEVIN RAYMOND BOYCE BUTLER

Professor	Address :	E301 CSE Building
Department of Computer & Information Science and Engineering		PO Box 116120
University of Florida		Gainesville FL 32611
email: butler@ufl.edu	Phone:	+1 (352) 294-2090
URL: http://www.kevinbutler.org	Fax:	+1 (352) 392-1220

Current Academic Appointment

Professor, Department of Computer and Information Science and Engineering, University of Florida, Aug. 2021 - *present*

Director, Florida Institute for Cybersecurity Research, University of Florida, Aug. 2022 - *present*

Educational Background

PH.D. in Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, August 2010 (Dissertation Title: *Leveraging Emerging Storage Functionality for New Security Services*; Advisor: Dr. Patrick McDaniel)

M.S. in Electrical Engineering, Columbia University, New York, NY, USA, May 2003 (Advisor: Dr. Henning Schulzrinne)

B.Sc. in Electrical Engineering, Queen's University, Kingston, ON, Canada, May 1999

Honors & Awards

Best Paper Award, ACM HotStorage, 2023.

Meta Research Award, Meta Research Labs, 2022-2023.

Best Paper Finalist, IEEE European Symposium on Security and Privacy, 2022.

University Term Professor, University of Florida, 2021-2024.

Best Paper Nominee, IEEE Conference on Virtual Reality, 2021.

Doctoral Dissertation Advisor/Mentoring Award, Herbert Wertheim College of Engineering, University of Florida, 2019.

Distinguished Poster Award, ACM CODASPY, 2019.

Arnold and Lisa Goldberg Rising Star Endowed Professorship, University of Florida, 2018-2024 (*relinquished in 2021 upon promotion to Professor*).

University Term Professor, University of Florida, 2018-2021.

Recognition of Service Award, Association for Computing Machinery, 2018.

Research Promotion Initiative Award, University of Florida, 2018.

Senior Member, Association for Computing Machinery, 2018.

International Educator of the Year, Herbert Wertheim College of Engineering, University of Florida, 2017.

Senior Member, IEEE, 2017.

Outstanding Community Service Award, IEEE Technical Committee on Security and Privacy, 2017.

Rising to National Preeminence Hire, University of Florida, 2014.

Best Teacher Award, Dept. of Computer and Information Science, University of Oregon, 2013.
CAREER Award, National Science Foundation, 2013.
Alumni Association Dissertation Award, Pennsylvania State University, 2010.
Symantec Intern Project Showcase Winner, Symantec Corp, 2009.
Graduate Research Assistant Award, Computer Science and Engineering Department, Pennsylvania State University, 2009.
Symantec Graduate Fellowship, Symantec Research Labs, 2009-2010.
Special Award of Meritous Service, Third International Conference on Information Systems Security, December 2007.
University Graduate Fellowship, Pennsylvania State University, 2006.
C. Norwood Wherry Memorial Graduate Fellowship, Pennsylvania State University, 2004.

Expert Witness Experience

During the course of my expert witness consulting engagements, I have written numerous expert reports, given six depositions, and have testified four times across two jury trials, with regards to both patent infringement and invalidity.

Testifying Expert, *Headwater Research LLC v. Samsung America Inc.*, No. 2-22-cv-00422, 2-22-cv-00467, 2-23-cv-00103 (E.D. Texas), retained by Fish and Richardson, representing defendant, April 2023 - present.

Testifying Expert, *Webroot, Inc. et al. v. Trend Micro Inc.*, No. 6:22-cv-00239-ADA-DTG (W.D. Texas), retained by Paul Hastings LLP and White & Case LLP, regarding intrusion detection devices, representing defendant. July 2022-present.

Consulting Expert, *Department of Justice, Antitrust Division, defendant confidential*, retained by USDOJ, regarding mobile devices. June 2022-June 2023.

Testifying Expert, *Research in Motion Limited ats Glen Snowball (File No. 13-57203CP), Blackberry Limited ats Michael Blackette (File No. 500-06-000583-118)*, retained by Torys LLP, regarding the Blackberry network, representing defendant. March 2021-present.

Testifying Expert, *IOENGINE LLC vs Roku Inc.* regarding portable devices, engaged by plaintiff in patent infringement case, October 2021-present.

Testifying Expert, *Finjian, Inc. v. Juniper Networks, Inc.* (3:17-cv-0569-WHA, District Court, N.D. California), retained by Irell and Manella LLP, regarding intrusion detection systems, representing defendant. June 2019-August 2019.

Testifying Expert, *Ingenico Inc v. IOENGINE LLC (1:18-cv-00826, District Court, D. Delaware)*, retained by Dechert LLP, regarding secure portable devices, representing defendant. October 2018-December 2022.

Testifying Expert, *IOENGINE LLC v. Imation Corp.* (1:14-cv-01572, District Court, D. Delaware), retained by Simpson Thacher & Bartlett LLP, regarding secure portable storage devices, representing plaintiff. March 2015-February 2017.

Testifying Expert, *IOENGINE LLC v. Interactive Media Corp.* (1:14-cv-01571, District Court, D. Delaware), retained by Simpson Thacher & Bartlett LLP, regarding secure portable storage devices, representing plaintiff. March 2015-January 2017.

Consulting Expert, *Vir2us, Inc.*, retained by Bunsow, De Mory, Smith, and Allison LLP, regarding virtualization and antivirus software. December 2014 - April 2015

Consulting Expert, *Envosys v. AT&T*, retained by Baker Botts, regarding location services in mo-

bile devices, representing defendant. October 2013-December 2013.

Research Support (\$36.7M total, \$9.34M investigator share)

Major Peer-Reviewed Grants (\$36.0M total, \$10.8M as PI, \$9.00M investigator share)

- **PI**, NSF CNS-2206950, “Collaborative Research: SaTC: Frontiers: Securing the Future of Computing for Marginalized and Vulnerable Populations”, National Science Foundation, \$7,500,000 (\$2,630,872 investigator share), 10/1/2022-9/30/2027.
- **Site PI**, NSF CNS-2054911, “Collaborative Research: SaTC: CORE: Medium: Enabling Practically Secure Cellular Infrastructure”, National Science Foundation, \$1,199,985 (\$598,019 investigator share), 1/1/2022-12/31/2024.
- **co-PI**, NSF CNS-2055123, “SaTC: CORE: Medium: Countering Surveillanceware Using Deception-Based Generative Models and Systems Mechanisms”, National Science Foundation, \$1,199,997 (\$618,531 investigator share), 1/1/2021-12/31/2024.
- **co-PI**, ONR N00014-20-1-2205, “Reconstructing Human Physiological Features from Audio Samples”, Office of Naval Research, \$700,000 (\$333,232 investigator share), 6/11/2021-6/10/2024.
- **co-PI**, DHS 70RSAT20CB0000017, “Deploying Defenses for Cellular Networks Using the AWARE Testbed”, DHS, \$3,122,033 (\$657,536 investigator share), 9/30/2020-9/29/2024.
- **PI**, ONR BAA-N00014-19-S-B001 Grant 12941701, “Mobile Interface Security Analysis”, Office of Naval Research, \$700,000 (\$376,242 investigator share), 1/1/2020-12/31/2022.
- **PI**, AFOSR AACOE, “Interpretable Security Reference Monitor for Deep Networks”, Air Force Office of Scientific Research, \$185,000 (\$185,000 investigator share), 12/1/2019-8/31/2021.
- **co-PI**, ONR DOD-19-STTR, “ArtusProtocol”, Office of Naval Research, \$800,000 (\$400,000 investigator share), 8/1/2019-7/31/2021.
- **co-PI**, AFOSR FA8650-19-1-1741, “University Center of Excellence: CYAN: Enabling Cyber Defense in Analog and Mixed Signal Domain”, Air Force Office of Scientific Research, \$5,000,000 (\$300,000 investigator share), 10/1/2019-9/30/2024. Selected for funding by AFOSR on 5/24/2019.
- **co-PI**, AFRL-FA9550-19-1-0169, “AFOSR University Center of Excellence in Assured Autonomy in Contested Environments (AACOE)”, Air Force Office of Scientific Research and Air Force Research Labs, \$5,999,515 (\$389,999 investigator share), 3/1/2019-2/28/2024.
- **co-PI**, DARPA HR00118S0052, “RACE: Resilient Communications for Everyone”, Defense Advanced Research Projects Agency, \$1,133,856 (\$382,698 investigator share), 3/1/2019-2/28/2023. Selected for funding by DARPA on 11/20/2018.
- **PI**, NSF/SRC CNS-1815883, “SaTC: STARSS: Small: Domain Informed Techniques for Detecting and Defending Against Malicious Firmware”, National Science Foundation, \$499,990 (\$250,000 investigator share), 11/01/2018-10/31/2021.

- **co-PI**, NSF CNS-1662976, “Collaborative Research: SURPASS: NSF SFS Unique Scholarship Program in Hardware and Systems Security”, National Science Foundation, \$2,508,541 (\$35,408 investigator share), 01/01/2017-12/31/2021.
- **co-PI**, NSF CNS-1562485, “TWC: Medium: Digital Healthcare-Associated Infection: Measurement, Defense and Prevention in a Modern Digital Healthcare Ecosystem”, National Science Foundation, \$1,200,000 (\$387,931 investigator share), 06/01/2016-05/31/2020.
- **PI**, “Developing Research Capability in Cyber-Physical Systems at the University of Florida”, Defense University Research Instrumentation Program (DURIP), Army Research Office, \$199,919 (\$100,000 investigator share), 09/01/2015-08/31/2016.
- **PI**, NSF CNS-1445983/1540216, “EAGER: Collaborative: Secure and Efficient Data Provenance”, National Science Foundation, \$205,000 (\$110,000 investigator share), 10/01/2014-03/31/2016.
- **PI**, NSF CNS 1254198/1540217, “CAREER: Securing Critical Infrastructure with Autonomously Secure Storage”, National Science Foundation, \$400,000, 04/01/2013-03/31/2018.
- **PI**, NSF CNS 1118046/154028, “TC:Small:Protection Mechanisms for Portable Storage”, National Science Foundation, \$515,530, 09/01/2011-08/31/2016.
- **co-PI**, “GAANN Program on Computer Systems Security”, US Department of Education, \$1,476,931 (\$184,616 investigator share), 08/16/2015-08/15/2018.
- **co-PI**, DARPA BAA 10-81, “Characterizing and Implementing Efficient Primitives for Privacy-Preserving Computation”, Defense Advanced Research Projects Agency and Air Force Research Labs, \$550,000 (\$262,375 investigator share), 05/01/2011-01/31/2015.

Industry and Other Support

- **co-PI**, “Mitigating Threat of Re-Identification from Eye-Tracking Data”, Meta Privacy-Enhancing Technology Award, \$100,000 ((\$49,000 investigator share)), 9/1/2022-8/31/2023.
- **co-PI**, “Center for Aerospace Resilience”, Florida Department of Education, \$875,000 (\$126,214 investigator share), 7/1/2020-7/31/2021.
- **PI**, “Secure Enclave Computing”, Royal Bank of Canada, \$87,347, 11/01/2019-01/01/2021.
- **PI**, NSF 1823067, “Travel Grant Support for ACM WiSec”, National Science Foundation, \$9,000, 05/27/2018-05/26/2019.
- **Co-PI**, “Formal Analysis of SGX Platform Software”, Intel Corporation, \$50,000 (\$25,000 investigator share), 01/01/2018-12/31/2018.
- **PI**, “Smartphones and Mobile money: Analysis and Recommendations for Security”, GSMA Mobile For Development Foundation, \$28,500 (\$13,500 investigator share), 12/01/2017-03/31/2018.
- **PI**, “Provenance-Based Enforcement Mechanisms”, MIT Lincoln Laboratory, \$137,000, 11/01/2013-10/31/2016.

- **PI**, “Securing Storage for Insider Threat Mitigation”, *Florida Cyber Consortium Seed Grant*, \$40,000 (\$20,000 investigator share), 2015.
- **co-PI**, “Resilience in BGPsec”, *Battelle*, \$180,000 \$80,000 investigator share, 02/01/2012-08/31/2013.
- **co-PI**, “A Cyber-Security Center of Excellence for Oregon”, *Oregon Engineering, Technology, and Industry Council*, \$180,000 (\$60,000 investigator share), 06/01/2014-05/31/2015.
- **co-PI**, “I3: An Interdisciplinary Approach to Internet Privacy”, *University of Oregon Research, Innovation and Graduate Education*, \$50,000, 04/10/2014-03/31/2015.
- **PI**, NSF CNS 1329307, “Oregon Security Day”, *National Science Foundation*, \$8,000, 03/01/2013-04/30/2014.
- **Recipient**, “USB Fingerprinting”, *Ellisys, equipment donation*, \$8,000, 2011.
- **Recipient**, “Project HAWAII”, *Microsoft Research, equipment donation*, \$1,000, 2011.

Publications (4986 citations [Google Scholar, July 2023], h-index=39, i10-index=73)

Book Chapters

1. Adam Bates, Devin Pohly, and Kevin R. B. Butler. “Secure and Trustworthy Provenance Collection for Digital Forensics.” In C. Wang, R. M. Gerges, Y. Guan, and S. K. Kasera, eds., *Digital Fingerprinting* (pp. 141–176). New York: Springer-Verlag, 2016.
2. Kevin Butler, William Enck, Patrick Traynor, Jennifer Plasterr, and Patrick McDaniel. “Privacy Preserving Web-Based Email”. *Algorithms, Architectures and Information Systems Security*, Statistical Science and Interdisciplinary Research. World Scientific Computing, November 2008.

Journal Publications

3. Brendan David-John, Kevin Butler, and Eakta Jain. “Privacy-preserving Datasets of Eye-tracking Samples with Applications in XR.” *IEEE Transactions on Visualization and Computer Graphics*, January 2023. 11 pages. *Selected for presentation at IEEE VR 2023 conference.*
4. Weidong Zhu and Kevin Butler. “NASA: NVM-Assisted Secure Deletion for Flash Memory.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, July 2022 (early access). Accepted for presentation at the International Conference on Embedded Software (EMSOFT) 2022.
5. Tuba Yavuz, Farhaan Fowze, Grant Hernandez, Ken (Yihang) Bai, Kevin Butler, and Dave (Jing) Tian. “ENCIDER: Detecting Timing and Cache Side Channels in SGX Enclaves and Cryptographic APIs.” *IEEE Transactions on Dependable and Secure Computing*, March 2022 (early access). 18 pages.
6. Brendan David-John, Diane Hosfelt, Kevin Butler, and Eakta Jain. “A Privacy-Preserving Approach to Streaming Eye-Tracking Data.” *IEEE Transactions on Visualization and Computer Graphics*, March 2021 (early access). 11 pages. *Best paper nominee at IEEE VR 2021 conference.*
7. Farhaan Fowze, Grant Hernandez, Dave (Jing) Tian, Kevin Butler, and Tuba Yavuz. “ProXray: Pro-

- tol Model Learning and Guided Firmware Analysis.” *IEEE Transactions on Software Engineering*, September 2019.
8. Grant Hernandez, Dave (Jing) Tian, Farhaan Fowze, Tuba Yavuz, Patrick Traynor, and Kevin Butler. “Towards Automated Firmware Analysis in the IoT Era.” *IEEE Security and Privacy*, 17(5), pg. 38-46, Sept.-Oct. 2019.
 9. Joseph Choi and Kevin Butler. “Secure Multiparty Computation and Trusted Hardware: Examining Adoption Challenges and Opportunities”. *Security and Communication Networks*, vol 2019, Article ID 1368905, 28 pages, April 2019.
 10. Benjamin Mood and Kevin Butler. “PAL: A System for Optimizing Secure Function Evaluation in Mobile Devices”. *Journal of Information Security and Applications*, 40(2018), pg. 78-91, March 2018.
 11. Bradley Reaves, Luis Vargas, Nolen Scaife, Dave (Jing) Tian, Logan Blue, Patrick Traynor, and Kevin Butler. “Characterizing the Security of the SMS Ecosystem with Public Gateways”. *ACM Transactions on Privacy and Security*, 22(1), Article 2, January 2018.
 12. Nolen Scaife, Patrick Traynor, and Kevin Butler. “Making Sense of the Ransomware Mess (and planning a sensible path forward)”. *IEEE Potentials*, 36(6), pg. 28-31, Nov/Dec. 2017.
 13. Dave (Jing) Tian, Kevin R. B. Butler, Joseph Choi, Patrick D. McDaniel, and Padma Krishnaswamy. “Securing ARP/NDP From the Ground Up.” *IEEE Transactions on Information Forensics and Security*, 12(9), pg. 2131-2143, September 2017.
 14. Adam Bates, Dave (Jing) Tian, Grant Hernandez, Thomas Moyer, Kevin R. B. Butler, and Trent Jaeger. “Taming the Costs of Trustworthy Provenance through Policy Reduction.” *ACM Transactions on Internet Technology*, 17(4), Article 34, September 2017.
 15. Bradley Reaves, Jasmine Bowers, Nolen Scaife, Adam Bates, Arnav Bhartiya, Patrick Traynor, and Kevin Butler. “Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications”. *ACM Transactions on Privacy and Security*, 20(3), Article 11, August 2017.
 16. Bradley Reaves, Jasmine Bowers, Sigmund Albert Gorski III, Rahul Bobate, Raymond Cho, Hiranava Sas, Sharique Hussain, Hamza Karachiwala, Nolen Scaife, Byron Wright, Kevin Butler, William Enck, and Patrick Traynor. “*droid: Assessment and Evaluation of Android Application Analysis Tools”. *ACM Computing Surveys*, 49(3), Article 55, November 2016.
 17. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. “Outsourcing Secure Two-Party Computation as a Black Box”. *Security and Communication Networks*, 9(14), pg. 2261-2275, September 2016.
 18. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. “Secure Outsourced Garbled Circuit Evaluation for Mobile Devices.” *Journal of Computer Security*, 24(2), pg. 137-180, 2016.
 19. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. “Accountable Wiretapping -or- I Know That You Can Hear Me Now”. *Journal of Computer Security*, 23 (2015), pg. 167-195, 2015.
 20. Adam Bates, Benjamin Mood, Joe Pletcher, Hannah Pruse, Masoud Valafar, and Kevin Butler. “On Detecting Co-Resident Cloud Instances Using Network Flow Watermarking Techniques”. *International Journal of Information Security*, 13(2), pg. 171-189, April 2014.
 21. Thomas Moyer, Kevin Butler, Joshua Schiffman, Patrick McDaniel, and Trent Jaeger. “Scalable Web

- Content Attestation”. *IEEE Transactions on Computers*, 61(5), pg. 686-699, May 2012.
22. Kevin Butler, Stephen McLaughlin, Thomas Moyer, and Patrick McDaniel. “New Security Architectures Based on Emerging Disk Functionality”. *IEEE Security and Privacy*, 8(5), pg. 34-31, Sept./Oct. 2010.
 23. Kevin Butler, Toni Farley, Patrick McDaniel, and Jennifer Rexford. “A Survey of BGP Security Issues and Solutions”. *Proceedings of the IEEE*, 98(1), pg. 100-122, January 2010.
 24. Patrick Traynor, Kevin Butler, William Enck, Kevin Borders, and Patrick McDaniel. “*malnets*: Large-Scale Malicious Networks via Compromised Wireless Access Points”. *Journal of Security and Communication Networks (SCN)*. 2009.
 25. Kevin Butler, Sunam Ryu, Patrick Traynor, and Patrick McDaniel. “Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems”. *IEEE Transactions on Parallel and Distributed Systems*, 20(12), pg. 1803-1815, December 2009.
 26. Patrick McDaniel, William Aiello, Kevin Butler, and John Ioannidis, Origin Authentication in Interdomain Routing. *Computer Networks*, 50(16), pg. 2953-2980, 14 November 2006.

Conference Publications

Top-4 Security Conferences

27. Daniel Olszewski, Allison Lu, Carson Stillman, Kevin Warren, Cole Kitroser, Alejandro Pascual, Divyajyoti, Kevin Butler, and Patrick Traynor. ““Get in Researchers; We’re Measuring Reproducibility”: A Reproducibility Study of machine Learning Papers in Tier 1 Security Conferences”. 30th ACM Conference on Computer and Communications Security (CCS’23), Copenhagen, Denmark, November 2023. *Acceptance rate TBD*.
28. Kyungtae Kim, Sungwoo Kim, Kevin Butler, Antonio Bianchi, and Dave (Jing) Tian. “Fuzz The Power: Dual-role State Guided Black-box Fuzzing for USB Power Delivery”. *32nd USENIX Security Symposium (Security’23)*, Anaheim, CA, USA, August 2023. *Acceptance rate TBD*.
29. Yazhou Tu, Liqun Shen, Md Imran Hossen, Sara Rampazzi, Kevin Butler, and Xiali Hei. “Auditory Eyesight: Demystifying μ s-Precision Keystroke Tracking Attacks on Unconstrained Keyboard Inputs”. *32nd USENIX Security Symposium (Security’23)*, Anaheim, CA, USA, August 2023. *Acceptance rate TBD*.
30. Yan Long, Pirouz Naghavi, Blas Kojusner, Kevin Butler, Sara Rampazzi, and Kevin Fu. “Side eye: Characterizing the Limits of POV Acoustic Eavesdropping from Smartphone Cameras with Rolling Shutters and Movable Lenses”. *44th IEEE Symposium on Security and Privacy (Oakland’23)*, San Francisco, CA, USA, May 2023. *Acceptance rate TBD*.
31. Tyler Tucker, Hunter Searle, Kevin Butler, and Patrick Traynor. “Blue’s Clues: Practical Discovery of Non-Discoverable Bluetooth Devices”. *44th IEEE Symposium on Security and Privacy (Oakland’23)*, San Francisco, CA, USA, May 2023. *Acceptance rate TBD*.
32. Logan Blue, Kevin Warren, Hadi Abdullah, Cassidy Gibson, Luis Vargas, Jessica O’Dell, Kevin Butler, and Patrick Traynor. “Who Are You (I Really Wanna Know)?” *31st USENIX Security Symposium (Security’22)*, Boston, MA, USA, Aug 2022. *Acceptance rate: 18.1%*.
33. Kyungtae Kim, Taegyu Kim, Ertza Warriach, Byoungyoung Lee, Kevin Butler, Antonio Bianchi,

- and Dave (Jing) Tian. “FuzzUSB: Hybrid Stateful Fuzzing of USB Gadget Stacks.” *43rd IEEE Symposium on Security and Privacy (Oakland’22)*, San Francisco, CA, USA, May 2022. *Acceptance rate: 14.5%.*
34. Grant Hernandez, Marius Muench, Dominik Maier, Alyssa Milburn, Shinjo Park, Tobias Scharnowski, Tyler Tucker, Patrick Traynor, and Kevin Butler. “FirmWire: Transparent Dynamic Analysis for Cellular Baseband Firmware.” *2022 ISOC Network and Distributed System Security Symposium (NDSS’22)*, San Diego, CA, USA, April 2022. *Acceptance rate: 16.2%.*
 35. Grant Hernandez, Dave (Jing) Tian, Anurag Yadav, Byron J. Williams, and Kevin Butler. “Big-MAC: Fine-Grained Policy Analysis of Android Firmware.” *29th USENIX Security Symposium (Security’20)*, Boston, MA, USA, August 2020. (*Acceptance rate: 16.1%.*)
 36. Dave (Jing) Tian, Grant Hernandez, Joseph Choi, Vanessa Frost, Peter Johnson, and Kevin Butler. “LBM: A Security Framework for Peripherals within the Linux Kernel.” *40th IEEE Symposium on Security and Privacy (Oakland’19)*, San Francisco, CA, USA, May 2019. (*Acceptance rate=12.0%.*)
 37. Luis Vargas, Logan Blue, Vanessa Frost, Christopher Patton, Nolen Scaife, Kevin Butler, and Patrick Traynor. “Digital Healthcare-Associated Infection Analysis of a Major Multi-Campus Hospital System.” *26th ISOC Network and Distributed System Security Symposium (NDSS’19)*, San Diego, CA, USA, February 2019. (*Acceptance rate=17.1%*)
 38. Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin Butler, and Joseph Wilson. “Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems.” *26th ISOC Network and Distributed System Security Symposium (NDSS’19)*, San Diego, CA, USA, February 2019. (*Acceptance rate=17.1%*)
 39. Luis Vargas, Gyan Hazarika, Rachel Culpepper, Kevin Butler, Thomas Shrimpton, Doug Szjada, and Patrick Traynor. “Mitigating Risk while Complying with Data Retention Laws.” *25th ACM Conference on Computer and Communications Security (CCS’18)*, Toronto, ON, Canada, October 2018. (*Acceptance rate=17.0%*)
 40. Dave (Jing) Tian, Grant Hernandez, Joseph Choi, Vanessa Frost, Christie Ruales, Patrick Traynor, Hayawardh Vijaykumar, Lee Harrison, Amir Rahmati, Mike Grace, and Kevin Butler. “ATtention Spanned: Comprehensive Vulnerability Analysis of AT Commands Within the Android Ecosystem.” *27th USENIX Security Symposium (Security’18)*, Baltimore, MD, USA, August 2018. (*Acceptance rate=19.1%*)
 41. Dave (Jing) Tian, Nolen Scaife, Deepak Kumar, Michael Bailey, Adam Bates, and Kevin Butler. “SoK: “Plug & Pray” Today – Understanding USB Insecurity in Versions 1 through C.” *2018 IEEE Symposium on Security and Privacy (Oakland’18)*, San Francisco, CA, USA, May 2018. *Nominated for Distinguished Paper Award. (Acceptance rate=11.5%)*
 42. Christian Peeters, Hadi Abdullah, Nolen Scaife, Jasmine Bowers, Patrick Traynor, Bradley Reaves, and Kevin Butler. “Sonar: Detecting SS7 Redirection Attacks Via Call Audio-Based Distance Bounding.” *2018 IEEE Symposium on Security and Privacy (Oakland’18)*, San Francisco, CA, USA, May 2018. (*Acceptance rate=11.5%*)
 43. Grant Hernandez, Farhaan Fowze, Dave (Jing) Tian, Tuba Yavuz, and Kevin Butler. “FirmUSB: Vetting USB Device Firmware using Domain Informed Symbolic Execution.” *24th ACM Conference on Computer and Communications Security (CCS’17)*, Dallas, TX, USA October 2017. (*Acceptance rate=18.1%.*)

44. Dave (Jing) Tian, Adam Bates, Kevin Butler, and Raju Rangaswami. "ProvUSB: Block-level Provenance-Based Data Protection for USB Storage Devices." *23rd ACM Conference on Computer and Communications Security (CCS'16)*, Vienna, Austria, October 2016. (Acceptance rate=16.5%.)
45. Dave (Jing) Tian, Nolen Scaife, Adam Bates, Kevin Butler, and Patrick Traynor. "Making USB Great Again with USBFILTER". *25th USENIX Security Symposium (Security'16)*, Austin, TX, USA, August 2016. (Acceptance rate=15.5%.)
46. Bradley Reaves, Dave Tian, Nolen Scaife, Logan Blue, Patrick Traynor, and Kevin Butler. "Sending out an SMS: Characterizing the Security of the SMS Ecosystem with Public Gateways". *2016 IEEE Symposium on Security and Privacy (Oakland'16)*, San Jose, CA, USA, May 2016. (Acceptance rate=13.3%.)
47. Adam Bates, Dave Tian, Kevin Butler, and Thomas Moyer. "Trustworthy Whole-System Provenance for the Linux Kernel". *24th USENIX Security Symposium (Security'15)*, Washington, DC, USA, August 2015. (Acceptance rate=15.7%)
48. Bradley Reaves, Nolen Scaife, Adam Bates, Patrick Traynor, and Kevin Butler. "Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World". *24th USENIX Security Symposium (Security'15)*, Washington, DC, USA, August 2015. (Acceptance rate=15.7%)
49. Benjamin Mood, Debayan Gupta, Kevin Butler, and Joan Feigenbaum. "Reuse It Or Lose It: More Efficient Secure Computation Through Reuse of Encrypted Values". *21st ACM Conference on Computer and Communications Security (CCS'14)*, Scottsdale, AZ, USA, November 2014. (Acceptance rate=19%)
50. Adam Bates, Joe Pletcher, Tyler Nichols, Braden Hollembaek, Dave Tian, Abdulrahman Alkhelaifi, and Kevin Butler. "Securing SSL Certificate Verification through Dynamic Linking". *21st ACM Conference on Computer and Communications Security (CCS'14)*, Scottsdale, AZ, USA, November 2014. (Acceptance rate=19%)
51. Adam Bates, Ryan Leonard, Hannah Pruse, Daniel Lowd, and Kevin Butler. "Leveraging USB to Establish Host Identity Using Commodity Devices." *21st ISOC Network and Distributed System Security Symposium (NDSS'14)*, San Diego, CA, USA, February 2014. (Acceptance rate=18.6%)
52. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. "Secure Outsourced Garbled Circuit Evaluation for Mobile Devices". *22nd USENIX Security Symposium (Security'13)*, Washington, DC, USA, August 2013. (Acceptance rate=16.2%.)
53. Benjamin Kreuter, ahbi shelat, Benjamin Mood, and Kevin Butler. "PCF: A Portable Circuit Format For Scalable Two-Party Secure Computation". *22nd USENIX Security Symposium (Security'13)*, Washington, DC, USA, August 2013. (Acceptance rate=16.2%.)
54. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. "Accountable Wiretapping -or- I Know That You Can Hear Me Now". *19th ISOC Network and Distributed System Security Symposium (NDSS'12)*, San Diego, CA, USA, February 2012. (Acceptance rate=17.6%)
55. Kevin Butler, Stephen McLaughlin, and Patrick McDaniel. Rootkit Resistant Disks. *15th ACM Conference on Computer and Communications Security (CCS'08)*, Alexandria, VA, USA, October 2008. (Acceptance rate=18.1%)
56. Patrick Traynor, Kevin Butler, William Enck, and Patrick McDaniel. Realizing Massive-Scale Con-

ditional Access Systems Through Attribute-Based Cryptosystems. *15th Annual Network and Distributed System Security Symposium (NDSS'08)*, San Diego, CA, USA. February 2008. (Acceptance rate=17.8%)

57. Kevin Butler, William Aiello, and Patrick McDaniel. Optimizing BGP Security by Exploiting Path Stability. *13th ACM Conference on Computer and Communications Security (CCS'06)*, Alexandria, VA, USA, November 2006. (Acceptance rate=14.8%)

Other Conference Publications

58. Christian Peeters, Tyler Tucker, Anushri Jain, Kevin Butler, and Patrick Traynor. "LeopardSeal: Detecting Call Interception via Audio Rogue Base Stations". *21st ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'23)*, Helsinki, Finland, June 2023. Acceptance rate: 20.7%
59. Washington Garcia, Pin-Yu Chen, Hamilton Scott Clouse, Somesh Jha, and Kevin Butler. "Less is More: Dimension Reduction Finds On-Manifold Adversarial Examples in Hard-Label Attacks". *IEEE Conference on Secure and Trustworthy Machine Learning (SatML 2023)*, Raleigh, NC, USA, February 2023. Acceptance rate: 25.3%
60. Daniel Olszewski, Weidong Zhu, Sandeep Sathyanarayana, Kevin Butler, and Patrick Traynor. "HallMonitor: A Framework for Identifying Network Policy Violations in Software". *2022 IEEE Conference on Communications and Network Security (CNS 2022)*, Austin, TX, USA, October 2022. (Acceptance rate TBD.)
61. Cassidy Gibson, Vanessa Frost, Katie Platt, Washington Garcia, Luis Vargas, Sara Rampazzi, Vincent Bindschaedler, Patrick Traynor, and Kevin Butler. "Analyzing the Monetization Ecosystem of Stalkerware." *22nd Privacy Enhancing Technologies Symposium (PETS 2022)*, Sydney, Australia, July 2022. (Acceptance rate: 24%)
62. Washington Garcia, Scott Clouse, and Kevin Butler. "Disentangling Categorization in Multi-agent Emergent Communication." *2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2022)*, Seattle, WA, USA, July 2022. Acceptance rate: 21%
63. Shantanu Ghosh, Zheng Feng, Kevin Butler, and Mattia Prosperi. "DR-VIDAL: Doubly Robust Variational Information-theoretic Deep Adversarial Learning for Counterfactual Prediction and Treatment Effect Estimation." *2022 American Medical Informatics Association Annual Symposium (AMIA 2022)*, Washington, DC, USA, November 2022.
64. Brendan David-John, Kevin Butler, and Eakta Jain. "For Your Eyes Only: Privacy-Preserving Eye-Tracking Datasets." *14th ACM Symposium on Eye Tracking Research and Applications (ETRA 2022)* Seattle, WA, USA, June 2022. (Acceptance rate: 42%)
65. Mounir Elbharabawy, Blas Kojusner, Mohammad Mannan, Kevin Butler, Byron Williams, and Amr Youssef. "SAUSAGE: Security Analysis of Unix domain Socket Usage in Android." *7th IEEE European Symposium on Security and Privacy (Euro S&P'22)*, Genoa, Italy, June 2022. (Acceptance rate TBD. Best paper finalist.)
66. Arslan Khan, Joseph Choi, Dave (Jing) Tian, Tyler Ward, Kevin Butler, Patrick Traynor, John M. Shea and Tan Wong. "Privacy-Preserving Localization Using Enclaves." *12th IEEE Ubiquitous Computing, Electronics and Mobile Communication Conference (IEEE UEMCON 2021)*, Virtual Confer-

ence, December 2021.

67. Washington Garcia, Animesh Chhotaray, Joseph Choi, Suman Adari, Kevin Butler, and Somesh Jha. “Brittle Features of Device Authentication.” *11th ACM Conference on Data and Application Security and Privacy (CODASPY’21)*, virtual event, April 2021. (Acceptance rate: 24.5%.)
68. Joseph Choi, Dave (Jing) Tian, Grant Hernandez, Christopher Patton, Benjamin Mood, Thomas Shrimpton, Patrick Traynor, and Kevin Butler. “A Hybrid Approach to Secure Function Evaluation using SGX.” *14th ACM ASIA Conference on Computer and Communications Security (ASIACCS’19)*, Auckland, New Zealand, July 2019. (Acceptance rate [full papers]: 17%.)
69. Vanessa Frost, Dave (Jing) Tian, Christie Ruales, Patrick Traynor, and Kevin Butler. “Examining DES-based Cipher Suite Support within the TLS Ecosystem.” *14th ACM ASIA Conference on Computer and Communications Security (ASIACCS’19)*, Auckland, New Zealand, July 2019. (Acceptance rate: 22%.)
70. Jasmine Bowers, Imani Sherman, Kevin Butler, and Patrick Traynor. “Characterizing Security and Privacy Practices in Emerging Digital Credit Applications.” *12th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec’19)*, Miami, FL, USA, May 2019. (Acceptance rate=25.5%)
71. Dave (Jing) Tian, Joseph Choi, Grant Hernandez, Patrick Traynor, and Kevin Butler. “A Practical Intel SGX Setting for Linux Containers in the Cloud.” *9th ACM Conference on Data and Application Security and Privacy (CODASPY’19)*, Dallas, TX, USA, March 2019. (Acceptance rate=23.5%)
72. Lianying Zhao, Joseph Choi, Didem Demirag, Kevin Butler, Mohammad Mannan, Erman Ayday, and Jeremy Clark. One-Time Programs Made Practical. *23rd International Conference on Financial Cryptography and Data Security (FC’19)*, St. Kitts, February 2019. (Acceptance rate=18.5%)
73. Robert Karam, Tamzidul Hoque, Kevin Butler, and Swarup Bhunia. Mixed-Granular Architectural Diversity for Device Security in the Internet of Things. *Asian Hardware Oriented Security and Trust Symposium (AsianHOST 2017)*, Beijing, China, October 2017.
74. Tyler Ward, Joseph Choi, Kevin Butler, John M. Shea, Patrick Traynor, and Tan Wong. Privacy Preserving Localization Using a Distributed Particle Filtering Protocol. *IEEE MILCOM*, Baltimore, MD, October 2017.
75. Jasmine Bowers, Bradley Reaves, Imani N. Sherman, Patrick Traynor, and Kevin Butler. Regulators, Mount Up! Analysis of Privacy Policies for Mobile Money Services. *13th Symposium on Usable Privacy and Security (SOUPS 2017)*, Santa Clara, CA, July 2017. (Acceptance rate=26.5%)
76. Adam Bates, Wajih Ul Hassan, Kevin Butler, Alin Dobra, Brad Reaves, Patrick Cable, Thomas Moyer, and Nabil Schear. Transparent Web Service Auditing via Network Provenance Functions. *26th World Wide Web Conference (WWW 2017)*, Perth, Australia, April 2017. (Acceptance rate=17.0%)
77. Benjamin Mood, Debayan Gupta, Henry Carter, Kevin Butler, and Patrick Traynor. “Frigate: A Validated, Extensible, and Efficient Compiler and Interpreter for Secure Computation”. *1st IEEE European Symposium on Security and Privacy (Euro S&P 2016)*, Saarbrücken, Germany, March 2016. (Acceptance rate=17.3%.)
78. Sriharsha Etigowni, Dave (Jing) Tian, Grant Hernandez, Saman Zonouz, and Kevin Butler. CPAC: Securing Critical Infrastructure with Cyber-Physical Access Control. *32nd Annual Computer Security Applications Conference*, Los Angeles, CA, USA, December 2016. (Acceptance rate=22.8%)

79. Thomas Moyer, Patrick Cable, Karishma Chadha, Robert Cunningham, Nabil Schear, Warren Smith, Adam Bates, Kevin Butler, Frank Capobianco, and Trent Jaeger. "Leveraging Data Provenance to Enhance Cyber Resilience". *1st IEEE Cybersecurity Development Conference (SecDev 2016)*, Boston, MA, November 2016. (Acceptance rate=52.2%.)
80. Bradley Reaves, Dave Tian, Nolen Scaife, Logan Blue, Patrick Traynor, and Kevin Butler. "Detecting SMS Spam in the Age of Legitimate Bulk Messaging". *9th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec'16)*, Darmstadt, Germany, July 2016. (Acceptance rate=35%.)
81. Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin Butler. "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data". *36th IEEE International Conference on Distributed Computing Systems (ICDCS 2016)*, Nara, Japan, June 2016. (Acceptance rate=17.6%)
82. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. "Outsourcing Secure Two-Party Computation as a Black Box". *14th International Conference on Cryptology and Network Security (CANS 2015)*, Marrakesh, Morocco, December 2015. (Acceptance rate=52.9%.)
83. Dave Tian, Adam Bates, and Kevin Butler. "Defending Against Malicious USB Firmware with GoodUSB". *31st Annual Computer Security Applications Conference (ACSAC 2015)*, Los Angeles, CA, USA, December 2015. (Acceptance rate=24.4%.)
84. Ethan Shernan, Henry Carter, Jing (Dave) Tian, Patrick Traynor, and Kevin Butler. "More Guidelines Than Rules: CSRF Vulnerabilities from Noncompliant OAuth 2.0 Implementations". *12th Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA 2015)*, Milano, Italy, July 2015. (Acceptance rate=22.7%)
85. Jing (Dave) Tian, Kevin Butler, Patrick McDaniel, and Padma Krishnaswamy. "Securing ARP From the Ground Up". *5th ACM Conference on Data and Application Security and Privacy (CODASPY 2015)*, San Antonio, TX, USA, March 2015. (Acceptance rate=33.3%)
86. Adam Bates, Joe Pletcher, Tyler Nichols, Braden Hollembaek, and Kevin Butler. "Forced Perspectives: Evaluating an SSL Trust Enhancement at Scale". *2014 ACM Internet Measurement Conference (IMC'14)*, Vancouver, BC, Canada, November 2014. (Acceptance rate=22.9%)
87. Adam Bates, Benjamin Mood, Masoud Valafar, and Kevin Butler. "Towards Secure Provenance-Based Access Control in Cloud Environments". *3rd ACM Conference on Data and Application Security and Privacy (CODASPY 2013)*, San Antonio, TX, USA, February 2013. (Acceptance rate=31.8%)
88. Vasant Tendulkar, Joe Pletcher, Ashwin Shashidharan, Ryan Snyder, Kevin Butler, and William Enck. "Abusing Cloud-based Browsers for Fun and Profit". *28th Annual Computer Security Applications Conference (ACSAC 2012)*, Orlando, FL, USA, December 2012. (Acceptance rate=19.0%)
89. Devin J. Pohly, Stephen McLaughlin, Patrick McDaniel, and Kevin Butler. "Hi-Fi: Collecting High-Fidelity Whole-System Provenance". *28th Annual Computer Security Applications Conference (ACSAC 2012)*, Orlando, FL, USA, December 2012. (Acceptance rate=19.0%)
90. Benjamin Mood, Lara Letaw, and Kevin Butler. "Memory-Efficient Garbled Circuit Generation for Mobile Devices". *16th IFCA International Conference on Financial Cryptography and Data Security (FC'12)*, Bonaire, February 2012. (Acceptance rate=26.1%)
91. Kevin Butler, Stephen McLaughlin, and Patrick McDaniel, "Kells: A Protection Framework for Portable Data". *26th Annual Computer Security Applications Conference (ACSAC 2010)*, Austin, TX,

USA, December 2010. (*Acceptance rate=16.3%*)

92. Machigar Ongtang, Kevin Butler, and Patrick McDaniel. "Porscha: Policy Oriented Secure Content Handling in Android." *26th Annual Computer Security Applications Conference (ACSAC 2010)*, Austin, TX, USA, December 2010. (*Acceptance rate=16.3%*)
93. Kevin Butler, Stephen McLaughlin, and Patrick McDaniel. "Disk-Enabled Authenticated Encryption" (short paper). *26th IEEE Symposium on Massive Storage Systems and Technologies (MSST 2010)*, Incline Village, NV, USA, May 2010. (*Acceptance rate=45.5%*.)
94. Thomas Moyer, Kevin Butler, Joshua Schiffman, Patrick McDaniel, and Trent Jaeger. Scalable Web Content Attestation. *25th Annual Computer Security Applications Conference (ACSAC 2009)*, Honolulu, HI, USA, December 2009. (*Acceptance rate=19.0%*.)
95. William Enck, Kevin Butler, Thomas Richardson, Patrick McDaniel, and Adam Smith. Defending Against Attacks on Main Memory Persistence. *24th Annual Computer Security Applications Conference (ACSAC 2008)*, Anaheim, CA, USA, December 2008. (*Acceptance rate=24.3%*)
96. Dhananjay Bapat, Kevin Butler, and Patrick McDaniel. Towards Automatic Privilege Separation. *3rd International Conference on Information Systems Security (ICISS 2007)*, Delhi, India, December 2007. (*Acceptance rate=25.0%*)
97. Lisa Johansen, Kevin Butler, Michael Rowell, and Patrick McDaniel. Email Communities of Interest. *Fourth Conference on Email and Anti-Spam (CEAS 2007)*, Mountain View, CA, USA, August 2007. (*Acceptance rate=19.0%*.)
98. Anusha Sriraman, Kevin Butler, Patrick McDaniel, and Padma Raghavan. Analysis of the IPv4 Address Space Delegation Structure. *IEEE Symposium on Computers and Communications (ISCC'07)*, Aveiro, Portugal, July 2007. (*Acceptance rate=40%*.)
99. Sunam Ryu, Kevin Butler, Patrick Traynor, and Patrick McDaniel. Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems. *3rd IEEE International Symposium on Security in Networks and Distributed Systems (SSNDS'07)*, Niagara Falls, ON, Canada, May 2007. (*Acceptance rate=40%*)
100. Kevin Butler, Patrick Traynor, William Enck, Jennifer Plasterr, and Patrick McDaniel. Privacy Preserving Web-Based Email. *2nd International Conference on Information Systems Security (ICISS 2006)*, Kolkata, India, December 2006. (*Acceptance rate=30.4%*)
101. Trent Jaeger, Kevin Butler, David King, Serge Hallyn, Joy Latten, and Xiolan Zhang. Leveraging IPsec for Mandatory Per-Packet Access Control. *2nd IEEE Communications Society/CreateNet International Conference on Security and Privacy in Communication Networks (SecureComm'06)*, August 2006, Baltimore, MD, USA. (*Acceptance rate=25.4%*.)
102. Kevin Butler and Patrick McDaniel, Understanding Mutable Internet Pathogens, or How I Learned to Stop Worrying and Love Parasitic Behavior. *1st International Conference on Information Systems Security (ICISS 2005)*, December 2005, Kolkata, India. (*Invited paper.*)

Workshop Publications

103. Jennifer Sheldon, Weidong Zhu, Adnan Abdulla, Kevin Butler, Md Jahidul Islam, and Sara Rappazzi. "Position: Deep Note: Can Acoustic Interference Damage the Availability of Hard Disk Storage

in Underwater Data Centers?” *15th ACM Workshop on Hot Topics in Storage and Files Systems (Hot-Storage 2023)*, Boston, MA, USA, July 2023. *Best Paper Award*.

104. Brendan David-John, Diane Hosfelt, Kevin Butler, and Eakta Jain. “Let’s SOUP Up XR: Collected thoughts from an IEEE VR workshop on privacy in mixed reality.” *1st International Workshop on Security for XR and XR for Security (VR4Sec)*, in conjunction with SOUPS 2021, virtual event, August 2021.
105. Sushrut Shringarputale, Patrick McDaniel, Kevin Butler, and Thomas La Porta. “Co-residency Attacks on Containers are Real.” *2020 ACM Cloud Computing Security Workshop (CCSW’20)*, Orlando, FL, USA, November 2020.
106. Suman Adari, Washington Garcia, and Kevin Butler. “Adversarial Video Captioning”. *2019 Dependable and Secure Machine Learning Workshop (DSML’19)*, Portland, OR, USA, June 2019.
107. Siddhant Deshmukh, Henry Carter, Grant Hernandez, Patrick Traynor, and Kevin Butler. “Efficient and Secure Template Blinding for Biometric Authentication”. *2nd IEEE Workshop on Security and Privacy in the Cloud (SPC 2016)*, Philadelphia, PA, USA, October 2016.
108. Debayan Gupta, Benjamin Mood, Joan Feigenbaum, Kevin Butler and Patrick Traynor. “Using Intel Software Guard Extensions for Efficient Two-Party Secure Function Evaluation.” *2016 Workshop on Encrypted Computing and Applied Homomorphic Cryptography (WAHC’16)*, Barbados, February 2016.
109. Adam Bates, Kevin Butler, and Thomas Moyer. “Take Only What You Need: Leveraging Mandatory Access Control Policy to Reduce Provenance Storage Costs.” *7th Annual Workshop on Theory and Practice of Provenance (TaPP’15)*, Edinburgh, Scotland, July 2015.
110. Adam Bates, Kevin Butler, Andreas Haeberlen, Micah Sherr, and Wenchao Zhou. “Let SDN Be Your Eyes: Secure Forensics in Data Center Networks.” *NDSS Workshop on Security of Emerging Network Technologies (SENT)*, San Diego, CA, USA, February 2014.
111. Matt Bishop, Emily Rine Butler, Kevin Butler, Carrie Gates, and Steven Greenspan. “Forgive and Forget: Return to Obscurity”. *New Security Paradigms Workshop (NSPW 2013)*, Banff, AB, Canada, September 2013.
112. Peter McKay, Bryan Clement, Sean Haverty, Elijah Newton, and Kevin Butler. “Read My Lips: Towards Use of the Microsoft Kinect as a Visual-Only Automatic Speech Recognizer.” *2013 Workshop on Home Usable Privacy and Security (HUPS)*, Newcastle, UK, July 2013.
113. Adam Bates, Benjamin Mood, Joe Pletcher, Hannah Pruse, Masoud Valafar, and Kevin Butler. “Detecting Co-Residency with Active Traffic Analysis Techniques”. *4th ACM Cloud Computing Security Workshop (CCSW 2012)*, Raleigh, NC, USA, October 2012. (*Acceptance rate=12.0% for full papers.*)
114. Lara Letaw, Joe Pletcher, and Kevin Butler. “Host Identification via USB Fingerprinting”. *6th IEEE International Workshop on Systematic Approaches to Digital Forensics Engineering (SADFE 2011)*, Oakland, CA, USA, May 2011. (*Acceptance rate=34.8%.*)
115. Kevin Butler, William Enck, Harri Hursti, Stephen McLaughlin, Patrick Traynor, and Patrick McDaniel. Systemic Issues in the Hart InterCivic and Premier Voting Systems: Reflections Following Project EVEREST. *2008 USENIX/ACCURATE Electronic Voting Workshop (EVT’08)*, July 2008. San Jose, CA, USA. (*Acceptance rate=44.1%*)
116. Shiva Chaitanya, Kevin Butler, Anand Sivasubramaniam, Patrick McDaniel and Murali Vilayan-

nur. Design, implementation and evaluation of security in iSCSI-based network storage systems. *2nd International Workshop on Storage Security and Survivability (StorageSS 2006)*, October 2006. Alexandria, VA, USA. (Acceptance rate=68%)

117. Patrick McDaniel, Kevin Butler, Stephen McLaughlin, Radu Sion, Erez Zadok, and Marianne Winslett. "Towards a Secure and Efficient System for End-to-End Provenance". *USENIX Workshop on Theory and Practice of Provenance (TaPP)*, February 2010. (Acceptance rate=68.8%)
118. Kevin Butler and Petros Efstathopoulos. U Can't Touch This: Block-Level Protection for Portable Storage. *International Workshop on Software Support for Portable Storage (IWSSPS 2009)*, October 2009. Grenoble, France.
119. Kevin Butler, Stephen McLaughlin, and Patrick McDaniel. Non-Volatile Memory and Disks: Avenues for Policy Architectures. *1st Computer Security Architectures Workshop (CSAW 2007)*, November 2007. Fairfax, VA, USA. (Acceptance rate=30%)
120. Kevin Butler and Patrick McDaniel, Testing Large Scale BGP Security in Replayable Network Environments. *DETER Community Workshop on Cyber Security Experimentation and Test*, June 2006. Arlington, VA, USA.
121. Sophie Qiu, Kevin Butler, and Patrick McDaniel, BGPRV: Retrieving and Processing BGP Data with Efficiency and Convenience. *DETER Community Workshop on Cyber Security Experimentation and Test*, June 2006. Arlington, VA, USA.

Technical Reports

122. Efe Bozkir, Süleyman Özdel, Mengdi Wang, Brendan David-John, Hong Gao, Kevin Butler, Eakta Jain, and Enkelejda Kasneci. "Eye-tracked Virtual Reality: A Comprehensive Survey on Methods and Privacy Challenges." *arXiv:2305.14080*, May 2023.
123. Washington Garcia, Pin-Yu Chen, Somesh Jha, Scott Clouse, and Kevin Butler. "Hard-label Manifolds: Unexpected Advantages of Query Efficiency for Finding On-manifold Adversarial Examples." *arXiv:2013.03325*, March 2021.
124. Kevin Butler. "Digital Financial Services security audit guideline." Security, Infrastructure and Trust Working Group, Financial Inclusion Global Initiative, published by International Telecommunications Union (ITU), March 2021.
125. Kevin Butler and Vijay Mauree. "Digital Financial Services security assurance framework." Security, Infrastructure and Trust Working Group, Financial Inclusion Global Initiative, published by International Telecommunications Union (ITU), January 2021.
126. Washington Garcia, Joseph Choi, Suman Adari, Somesh Jha, and Kevin Butler. "Explainable Black-Box Attacks Against Model-based Authentication." *arXiv:1810.00024*, September 2018.
127. Kevin Butler, Patrick Traynor, and Tiago Novais. "MM App Security Best Practices." Official Document MM.01, GSM Association, London, United Kingdom, June 2018.
128. Kevin Butler, Leon Perlman, Paul Makin, Henry Gerwitz, Patrick Traynor, Yury Grin, Evgeniy Bondarenko, and Richard Miller. "ITU-T Focus Group Digital Financial Services: Security Aspects of Digital Financial Services (DFS)." Focus Group Technical Report, International Telecommunications Union, Geneva, Switzerland, January 2017.

129. Adam Bates, Joe Pletcher, Tyler Nichols, Braden Hollembaek, and Kevin Butler. "Towards Usable System-Wide Trust Agility." Technical Report CIS-TR-2013-13, Department of Computer and Information Science, University of Oregon, Eugene, OR, October 2013.
130. Benjamin Mood and Kevin Butler. "Optimizing Secure Function Evaluation for Mobile Devices". Technical Report CIS-TR-2013-10, Department of Computer and Information Science, University of Oregon, Eugene, OR, September 2013.
131. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. "Secure Outsourced Garbled Circuit Evaluation for Mobile Devices". Technical Report GT-CS-12-09, College of Computing, Georgia Institute of Technology, Atlanta, GA, December 2012.
132. Kevin Butler and Petros Efstathopoulos. U Can't Touch This: Block-Level Protection for Portable Storage. Technical Report Symantec-SRL/MV2009-11, Symantec Research Labs, Mountain View, CA, June 2009.
133. Kevin Butler, Stephen McLaughlin, Thomas Moyer, Joshua Schiffman, Patrick McDaniel, and Trent Jaeger. Firma: Disk-Based Foundations for Trusted Operating Systems. Technical Report NAS-TR-0114-2009, Networking and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, May 2009.
134. Kevin Butler, Stephen McLaughlin, Thomas Moyer, Patrick McDaniel, and Trent Jaeger. Switch-Blade: Policy-Driven Disk Segmentation. Technical Report NAS-TR-0098-2008, Networking and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, November 2008.
135. P. McDaniel, K. Butler, W. Enck, H. Hursti, S. McLaughlin, P. Traynor, M. Blaze, A. Aviv, P. Cerny, S. Clark, E. Cronin, G. Shah, M. Sherr, G. Vigna, R. Kemmerer, D. Balzarotti, G. Banks, M. Cova, V. Felmetsger, W. Robertson, F. Valeur, J. Hall, and L. Quilter, EVEREST: Evaluation and Validation of Election-Related Equipment, Standards and Testing. Public Report, Ohio Secretary of State, December 2007.
136. Kevin Butler, Stephen McLaughlin, Patrick McDaniel, and Youngjae Kim. Autonomously Secure Disks. Technical Report NAS-TR-0072-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, May 2007. (*Updated September 2007.*)
137. Lisa Johansen, Kevin Butler, William Enck, Patrick Traynor, and Patrick McDaniel. Grains of SANs: Building Storage Area Networks from Memory Spots. Technical Report NAS-TR-0060-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, January 2007.
138. Luke St. Clair, Lisa Johansen, Kevin Butler, William Enck, Matthew Pirretti, Patrick Traynor, Patrick McDainel, and Trent Jaeger. Password Exhaustion: Predicting the End of Password Usefulness. Technical Report NAS-TR-0030-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, May 2006. (*Updated January 2007.*)

Articles for Media

139. Patrick Traynor and Kevin Butler. "Mobile money in developing countries: study reveals secu-

urity flaws in apps”. *The Guardian*. 24 September 2015. http://www.theguardian.com/global-development-professionals-network/2015/sep/24/mobile-money-apps-security-flaws-study-reveals?CMP=tw_t_gu

Patents

U.S. Patent 8,856,918, Host validation mechanism for preserving integrity of portable storage data. Petros Efstathopoulos, Bruce Montague, Dharmesh Shah, Kevin Butler. October 7, 2014.

U.S. Patent 10,685,114, Malware Detection Via Data Transformation Monitoring. Walter N. Scaife, Kevin Butler, Henry Carter, Patrick G. Traynor. June 16, 2020.

U.S. Patent 11,265,717, Detecting SS7 Redirection Attacks with Audio-Based Distance Bounding. Patrick G. Traynor, Christian Peeters, Bradley G. Reeves, Hadi Abdullah, Kevin Butler, Jasmine Bowers, Walter N. Scaife. March 1, 2022.

U.S. Patent 11,568,044, Method and Apparatus For Vetting Universal Serial Bus Device Firmware. Kevin Butler, Tuba Yavuz, Jing Tian, Grant Hernandez, Farhaan Fowze. January 31, 2023.

U.S. Patent 11,640,464, Protocol Model Learning and Guided Firmware Analysis. Tuba Yavuz, Farhaan Fowze, Kevin Raymond Boyce Butler, Jing Tian, Grant Haydock Hernandez. May 2, 2023.

U.S. Patent 11,663,388, Automated Security Analysis of Baseband Firmware. Grant Haycock Hernandez, Kevin R. Butler, Patrick G. Traynor. May 31, 2023.

Teaching

University of Florida

- *CIS 4360: Computer and Information Security*, undergraduate, Fall 2020, Fall 2019, Fall 2018, Fall 2017.
- *CIS 5370: Computer and Information Security*, graduate, Spring 2023, Spring 2019, Spring 2018, Spring 2017.
- *CIS 6930: Special Topics: Mobile and Embedded System Security*, graduate, Spring 2020.
- *CNT 5410: Computer and Network Security*, graduate, Fall 2015.

University of Oregon

- *CIS 415: Operating Systems (substantially revised)*, Spring 2014, Spring 2013, Spring 2012, Spring 2011.
- *CIS 607: Seminar on Generating Trustworthy Data (new)*, Winter 2014.
- *CIS 607: Seminar on Embedded Systems and Security (new)*, Spring 2013.
- *CIS 433/533: Computer and Network Security*, Winter 2013, Winter 2011.
- *CIS 607: Seminar on Computer Security in the Physical World (new)*, Spring 2012.
- *CIS 330: C/C++ and UNIX (substantially revised)*, Winter 2012.
- *CIS 610: Advanced Topics in Systems Security (new)*, Fall 2011.
- *CIS 607: Seminar on Security in Systems, Storage, and Clouds (new)*, Fall 2010.

The Pennsylvania State University

- *EE/CSE 458: Computer Networks*, Summer 2006.

Professional Service

Editorial Positions, Panels, and Boards

Editorial Boards:

Associate Editor, ACM Transactions on Security and Privacy (TOPS), 2021-present.

Associate Editor, Journal of Computer Security, 2020-present.

Associate Editor, IEEE Transactions on Dependable and Secure Computing (*Impact Factor: 4.41*), 2019-2023.

Associate Editor, ACM Digital Threats: Research and Practice, 2017-2022.

Editorial Board, International Journal of Cybersecurity Science and Technology, 2017-2022.

Review Board, IEEE Systems Journal, Special Issues on Security and Privacy in Complex Systems, 2012.

Scientific Grant Panels:

Review Panelist, National Science Foundation, 2022, 2021, 2020, 2018, 2017, 2016, 2014 (2), 2013, 2012.

Review Panelist, American Association for the Advancement of Science (AAAS), 2019.

Grant Reviewer, Austrian Science Fund, 2018.

Grant Reviewer, Natural Science and Engineering Research Council of Canada, 2022, 2019, 2018, 2017.

Grant Reviewer, Israel Science Foundation, 2016.

Conference Steering Committees:

Steering Committee Member, ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2019 -*present*.

Steering Committee Member, Annual Computer Security Applications Conference, 2014 -*present*.

Service to National and International Organizations:

Council Member, Computing Research Association Computing Community Consortum, 2023-present.

Co-Chair, Security, Infrastructure, and Trust Working Group, Financial Inclusion Global Initiative, International Telecommunications Union, 2017-2022.

Co-Chair and Leader of Security Workstream, Technology, Innovation, and Competition Working Group, Focus Group on Digital Financial Services, International Telecommunications Union, 2015-2016.

Conference & Workshop Chairships

1. **Technical Program Co-Chair**, 32nd USENIX Security Symposium (USENIX Security'22), Boston, MA, USA, August 2022.
2. **Conference General Chair**, 36th and 37th Annual Computer Security Applications Conference (ACSAC'20, ACSAC'21), Austin, TX, USA, 2020,2021.
3. **Technical Program Co-Chair**, 16th EAI International Conference on Security and Privacy in Communication Networks (Securecomm 2020), Washington, DC, USA, October 2020.
4. **Technical Program Co-Chair**, 2020 USENIX Summit on Hot Topics in Security (HotSec'20), Boston, MA, USA, August 2020 (*suspended due to COVID-19 pandemic*)
5. **Technical Program Co-Chair**, 11th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec'18), Stockholm, Sweden, July 2018.
6. **Conference General Chair**, 38th IEEE Symposium on Security and Privacy, San Jose, CA, May 2017.
7. **Vice Chair and Registration Chair**, 37th IEEE Symposium on Security and Privacy, San Jose, CA, May 2016.
8. **Technical Program Chair**, 30th Annual Computer Security Applications Conference (ACSAC 2014), New Orleans, LA, December 2014.
9. **Technical Program Co-Chair**, 29th Annual Computer Security Applications Conference (ACSAC 2013), New Orleans, LA, December 2013.

Other Conference & Workshop Organization Activities (118 Tech. Program Committees)

Top-4 Security Conferences

10. **USENIX Security Symposium**: *TPC Member*, 2021, 2020, 2019, 2018, 2017, 2015, 2014; *Short Talks Co-Chair*, 2017.
11. **IEEE Symposium on Security and Privacy (Oakland)**: *TPC Member*, 2024, 2023, 2021, 2020, 2018, 2017, 2016, 2012; *Publications Chair*, 2014, 2013, 2012, 2011, *Submissions Chair*, 2008.
12. **ISOC Network and Distributed Security Symposium (NDSS)**: *TPC Member*, 2021, 2020, 2019, 2018, 2017, 2016.
13. **ACM Conference on Computer and Communications Security (CCS)**: *TPC Member*, 2020, 2016, 2015, 2014, 2013; *TPC Member (Industry Track)*, 2007, 2004.

Other Conferences and Workshops

14. **Privacy Enhancing Technologies Symposium (PETS)/Journal of Privacy Enhancing Technologies**, *Senior PC Member*, 2023, *TPC Member/Editorial Board*, 2014-2016.
15. **IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)**: *TPC Member*, 2023.
16. **ACM Conference on Security in Wireless and Mobile Networks (WiSec)**: *TPC Member*, 2023, 2022, 2021, 2020, 2019, 2017, 2016, 2015, 2014.
17. **IEEE Secure Development Conference (SecDev)**: *TPC Member*, 2023, 2022.
18. **USENIX Annual Technical Conference (ATC)**: *TPC Member*, 2021, 2020.
19. **IEEE European Symposium on Security and Privacy (Euro S&P)**: *TPC Member*, 2020.
20. **USENIX Workshop on Cyber Security Experimentation and Test (CSET)**: *TPC Member*, 2019, 2015, 2014, 2013.
21. **International Workshop on the Theory and Practice of Provenance (TaPP)**: *TPC Member*, 2019.
22. **USENIX Enigma Conference**: *TPC Member*, 2019, 2018.
23. **ACM Cloud Computing Security Workshop (CCSW)**: *TPC Member*, 2019, 2014.
24. **International Provenance and Annotation Workshop (IPAW)**: *TPC Member*, 2018.
25. **ACM Symposium on Information, Computer and Communications Security (ASIACCS)**: *TPC Member*, 2018, 2017, 2015, 2014.
26. **Annual Computer Security Applications Conference (ACSAC)**: *TPC Member*, 2018, 2017, 2012, 2011; *Test of Time Awards Chair*, 2019; *Panels Chair*, 2012, 2011; *Publicity Chair*, 2010.
27. **World Conference on Information Security Applications (WISA)**: *TPC Member*, 2017.
28. **International Conference on Cryptology and Network Security (CANS)**: *TPC Member*, 2017.
29. **New Security Paradigms Workshop (NSPW)**: *TPC Member*, 2016, 2015, 2014, 2013.
30. **IEEE Mobile Security Technologies Workshop (MOST)**: *TPC Member*, 2015, 2014.
31. **Symposium and Bootcamp on the Science of Security (HotSoS)**: *TPC Member*, 2015.
32. **Workshop on Encrypted Computing and Applied Homomorphic Cryptography (WAHC)**: *TPC Member*, 2015.
33. **European Symposium on Research in Computer Security (ESORICS)**: *TPC Member*, 2014, 2013, 2012.
34. **Smart Energy Grid Security Workshop (SEGS)**: *TPC Member*, 2014, 2013.
35. **Conference on Privacy, Security and Trust (PST)**: *TPC Member*, 2014, 2013.
36. **Global Internet Symposium (GI)**: *TPC Member*, 2014.
37. **IEEE International Conference on Big Data (BigData)**: *TPC Member*, 2013.
38. **IEEE LCN Network Security Workshop (WNS)**: *TPC Member*, 2013.
39. **International Workshop on Security in Information Systems (WOSIS)**: *TPC Member*, 2013,

2012.

40. **International Conference on Applied Cryptography and Network Security (ACNS):** *TPC Member*, 2013.
41. **International Workshop on Security (IWSEC):**, *TPC Member*, 2012, 2008, 2007.
42. **Workshop on Secure Network Protocols (NPSec):** *Publicity Chair*, 2012.
43. **LCN Workshop on Security in Communication Networks (SICK):** *TPC Member*, 2012.
44. **USENIX Workshop on Hot Topics in Security (HotSec):** *TPC Member*, 2012.
45. **International Workshop on Security (IWSEC):** *TPC Member*, 2011, 2010.
46. **International Conference on Security and Privacy in Communication Networks (SecureComm):** *TPC Member*, 2011.
47. **International Conference on Availability, Reliability and Security (ARES):** *TPC Member*, 2011, 2010, 2009, 2008, 2007.
48. **International Conference on Internet Monitoring and Protection (ICIMP):** *TPC Member*, 2011, 2010.
49. **International Conference on Financial Cryptography and Data Security (FC):** *TPC Member*, 2011.
50. **International Conference on Information Systems Security (ICISS):** *TPC Member*, 2009, 2008, 2007; *Submissions and Web Chair*, 2007.
51. **IEEE International Workshop on Software Security Process:** *TPC Member*, 2009.
52. **International Conference on International Conference on Information Security and Assurance (ISA):** *TPC Member*, 2009, 2008; *Publicity Co-Chair*, 2008.
53. **International Workshop on Security in Systems and Networks (SSN):** *TPC Member*, 2009, 2007.
54. **Conference on Future Generation Communication and Networking (FGCN):** *TPC Member*, 2008, 2007.
55. **IEEE International Workshop on Security in Software Engineering (IWSSE):** *TPC Member*, 2008, 2007.
56. **IEEE/IFIP Conference on Dependable Systems and Networks (DSN):** *TPC Member (Fast Abstracts)*, 2008.
57. **International Workshop on Secure Software Engineering (SecSE):** *TPC Member*, 2008.

Conference Session Chair: USENIX Security'18, ACSAC'17, WWW'17, IEEE SP'16, CCS'16, CCS'14, HotSec'12, IEEE SP'12, CCS'09

Journal & Book Reviewing

IEEE Transactions on Dependable and Secure Computing (TDSC), 2019, 2014, 2010; Communications of the ACM (CACM), 2019; IEEE Transactions on Computers (TC), 2019, 2012, 2009; IEEE Security and Privacy, 2018, 2013, 2010; Journal of Computer Security (JCS), 2018; IEEE Transactions on Information Forensics and Security (TIFS), 2017, 2013, 2011; Journal of Internet Services and

Applications (JISA), 2017; ACM Transactions on Information and System Security (TISSEC), 2016, 2015, 2014, 2010, 2009, 2008, 2006; IEEE Transactions on Reliability (TR), 2014; ACM Transactions on Internet Technologies (TOIT), 2013, 2010, 2009; ACM Computing Surveys, 2013; ACM Transactions on Storage (TOS), 2013, 2011; Elsevier Computers and Security, 2013, 2011; International Journal of Computer Mathematics, 2011; IEEE Transactions on Parallel and Distributed Systems (TPDS), 2011, 2008; Wiley and Sons (Book proposal reviewer), 2010; EURASIP Journal on Information Security (JIS), 2011, 2010; Elsevier Journal of Computer Networks, 2010, 2005; International Journal of Software and Informatics (IJSI), 2010; International Journal of Security and Networks (IJSN), 2009, 2007; The Computer Journal (Oxford), 2009; ACM/IEEE Transactions on Networking (TON), 2008; Wireless Personal Communications, 2008; IEEE Communications Surveys and Tutorials, 2007; IEEE Transactions on Software Engineering (TSE), 2007; Software: Practice and Experience, 2007; Handbook of Information Security, 2004.

Conference & Workshop Reviewing

USENIX OSDI, 2016; IEEE INFOCOM, 2014, 2007; IEEE Symposium on Security and Privacy (S&P), 2013, 2009, 2008, 2007; ISOC Symposium on Network and Distributed Systems Security (NDSS), 2013, 2012, 2009, 2007; ACM Conference on Data and Application Security and Privacy (CODASPY), 2013; IEEE Computer Security Foundations Symposium (CSF), 2011; ACM Symposium on Computer and Communications Security (CCS), 2010, 2009, 2008, 2006; Annual Computer Security Applications Conference (ACSAC), 2010, 2007, 2006, 2005; USENIX Workshop on Hot Topics in Security (HotSec), 2010, 2007; ACM Symposium on Access Control Models and Technologies (SACMAT), 2010, 2009, 2008, 2006; IFCA International Conference on Financial Cryptography and Data Security (FC), 2010, 2008, 2007; Information Security Conference (ISC), 2009; USENIX Security Symposium, 2009, 2008, 2007, 2006, 2004; IEEE GLOBECOM, 2008; IFIP Conference on Data and Applications Security (DBSEC), 2008, 2006; USENIX Conference on File and Storage Technologies (FAST), 2008; IEEE International Conference on Computer Engineering and Systems (ICCES), 2007; IEEE International Conference on Distributed Computing Systems (ICDCS), 2007; IEEE Sarnoff Symposium, 2007; IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2007; ACM International Workshop on Storage Security and Survivability (StorageSS), 2006; International Workshop on Security (IWSEC), 2006; International Conference on Emerging Trends in Computer Science (ETRICS), 2006; International Conference on Applied Cryptography and Network Security (ACNS), 2006; European Symposium on Research in Computer Security (ESORICS), 2005.

University Service

Sabbatical Committee, Member, College of Engineering, University of Florida, 2021-present.

Steering Committee, Member, University of Florida, 2018-present.

Graduate Admissions Committee, Member, CISE, University of Florida, 2019-present.

Consortium for Media and Trust Director Search Committee, Member, University of Florida, 2019.

Faculty Search Committee, Chair, CISE, University of Florida, 2018-2021.

Faculty Search Committee, Member, CISE, University of Florida, 2017-2018.

Awards Committee, Member, CISE, University of Florida, 2015-present.

Graduate Affairs Committee, Member, CISE, University of Florida, 2014-2018.

Graduate Education Committee, Member, CIS, University of Oregon, 2010-2014.

Undergraduate Curriculum Committee, Member, CIS, University of Oregon, 2012-2014.

General Education Committee, Member, University of Oregon, 2013-2014.

Students

Graduate Advisees

Past Ph.D Advisees:

- Washington Garcia, Fall 2022, now **Research Scientist**, University of Dayton Research Institute.
- Grant Hernandez, Summer 2020, now **Senior Security Engineer**, Qualcomm.
- Dave (Jing) Tian, Summer 2019, now **Assistant Professor of Computer Science**, Purdue University (NSF CAREER award, 2022).
- Adam Bates, Summer 2016, now **Associate Professor of Computer Science**, University of Illinois at Urbana-Champaign (NSF CAREER award, 2018).
- Benjamin Mood Spring 2016, now **Assistant Professor of Computer Science**, Point Loma Nazarene University.

Current Ph.D. Advisees: Yunhao Wu (PhD, Spring 2024), Weidong Zhu (PhD, Spring 2024, co-advised with Sara Rampazzi), Cassidy Gibson (PhD, Spring 2025, co-advised with Patrick Traynor), Anna Crowder (PhD, Spring 2026), Caroline Fedele (PhD, Spring 2026), Magdalena Pasternak (PhD, Spring 2026), Carson Stillman (PhD, Spring 2028), Kevin Childs (PhD, Spring 2030).

Past Master's Degree Students: Benjamin Simon (MS, Spring 2023), Reza Shahriari (MS, Fall 2023), Blas Kojusner (MS, Fall 2022), Hunter Searle (MS, Summer 2022), Vanessa Frost (MS, Spring 2022), Joseph Choi (MS, Spring 2021), Joseph Pletcher (MS, Spring 2013), Adam Bates (MS, Spring 2012), Benjamin Mood (MS, Spring 2012), Masoud Valafar (MS, Fall 2011).

Other Graduate Student Advising

PhD Committee Member: Ghulam Memon (PhD, University of Oregon, Fall 2013), Saeed Sadeghian (PhD, University of Calgary, Fall 2015), Henry Carter (PhD, Georgia Institute of Technology, Spring 2016), Mehdi Sadi (PhD, University of Florida, Spring 2017), Bradley Reaves (PhD, University of

Florida, Summer 2017), Robert Karam (PhD, University of Florida, Summer 2017), Luka Malisa (PhD, ETH Zurich, Summer 2017), Fengchao Zhang (PhD, University of Florida, Spring 2018), Kai Yang (PhD, University of Florida, Summer 2018), Nolen Scaife (PhD, University of Florida, Spring 2019), Ruimin Sun (PhD, University of Florida, Spring 2019), Francesco Pittaluga (PhD, University of Florida, Spring 2019), Jonathan Smith (PhD, University of Florida, Spring 2020), Beong Heyn Kim (PhD, University of Toronto, Spring 2020), Christopher Patton (PhD, University of Florida, Summer 2020), Xiaoyong Yuan (PhD, University of Florida, Summer 2020), Farhaan Fowze (PhD, University of Florida, Summer 2020), Tamzidul Hoque (PhD, University of Florida, Summer 2020), Jasmine Bowers (PhD, University of Florida, Summer 2020), Olufemi Odegbile (PhD, University of Florida, Spring 2021), Luis Vargas (PhD, University of Florida, Spring 2021), Atul Prasad Deb Nath (PhD, University of Florida, Spring 2021), Abdulrahman Alaqi (PhD, University of Florida, Spring 2021), Aritra Dhar (PhD, ETH Zurich, Summer 2021), Olufemi Odegbile (PhD, University of Florida, Summer 2021), Yazhou Tu (PhD, University of Louisiana at Lafayette, Fall 2021), Shubhra Deb Paul (PhD, University of Florida, Spring 2022), Hadi Abdullah (PhD, University of Florida, Spring 2022), Christian Peeters (PhD, University of Florida, Summer 2022), Brendan David-John (University of Florida, Summer 2022), Sarah Amir (PhD, University of Florida, Summer 2022), Jonathan Cruz (PhD, University of Florida, Fall 2022), Logan Blue (PhD, University of Florida, Spring 2023), Animesh Chhotaray (PhD, University of Florida, Spring 2023), Martin Georgiev (PhD, University of Oxford, Spring 2023), Sujun Kumar Saha (PhD, University of Florida, Spring 2023), Minh Vu (PhD, University of Florida, Spring 2024), Chaoyi Ma (PhD, University of Florida, Spring 2023), Yihang (Ken) Bai (PhD, University of Florida, Spring 2024), Anurag Swarnim Yadav (PhD, University of Florida, Spring 2024), Manish Merugu (PhD, University of Florida, Spring 2025), Yichen Jian (PhD, University of Florida, Spring 2025), Christopher Brant (PhD, University of Florida, Spring 2025), Mohammad Sami Ul Islam Sami (PhD, University of Florida, Spring 2025), Seth Layton (PhD, University of Florida, Spring 2026), Ethan Wilson (PhD, University of Florida, Spring 2026), Wenxuan Bao (PhD, University of Florida, Spring 2026).

Other Committee Membership: Ming Liu (University of Florida, thesis proposal committee, Spring 2016), Reza Motamedi (University of Oregon, thesis proposal committee, Spring 2014), Jason Gustafson (University of Oregon, MS Committee, Winter 2012), Amir Amir Farzad (University of Oregon, MS committee, Spring 2013).

Undergraduate Students

Hannah Pruse (Honors BS, Oregon, Spring 2013 - **NSF Graduate Fellow**), Ryan Leonard (Honors BS, Oregon, Spring 2013), Peter McKay (Honors BS, Oregon, Spring 2013), Braden Hollembaek (Honors BS, Oregon, Spring 2014), Abdul Alkhelaifi (Honors BS, Oregon Spring 2014), Devan Patel (BS, Florida, Spring 2015), Cynthia Omauzo (CS, Prairie View A&M, CRA DREU, hosted Summer 2015), Jared Trinkler (BS, University Scholar, Florida, Spring 2017), Joshua Uduehi (CRA DREU, BS, Indiana University, hosted Summer 2017), Suman Adari (BS, Florida, Spring 2021), Christie Ruales (BS, CRA/ACSA SWSIS Scholar, University Scholar, Florida, Spring 2021), César Arguello (BS, Florida, Spring 2022), Katie Platt (BS, Florida, Fall 2022), Gabriella Neris (BS, Florida, Spring 2023).

Presentations & Invited Talks

1. **Keynote Address:** From Blue Boxes to Black Boxes: Adventures in Uncovering Device Function-

- ality. *IEEE Workshop on Offensive Technology (WOOT)*, San Francisco, CA, May 2013.
2. From Blue Boxes to Black Boxes: Telecommunications Security Research at the Florida Institute for Cybersecurity Research. *Florida Cybersecurity Advisory Council*, Gainesville, FL, May 2013.
 3. Privacy-Preserving Computation for Resource-Constrained Devices. *NATO Science and Technology Office ET Meeting*, Washington, DC, November 2022.
 4. Investigating External Surfaces in Android Devices. *Microsoft Research*, Cambridge, United Kingdom, February 2020.
 5. Firmware-Informed Approaches to Securing Systems and Devices. *ARM Research*, Cambridge, United Kingdom, February 2020.
 6. Cybersecurity and Election Security: Innovations in Research and Technology (with Juan Gilbert). *Orange and Blue Plate Luncheon*, Tallahassee, FL, February 2020.
 7. Making MPC More Accessible Through Informed Compiler Design. *Workshop on Teaching Secure Computation*, Washington, DC, January 2020.
 8. Analyzing Applications to Assess Security. *FIGI Security Clinic*, Geneva, Switzerland, December 2019.
 9. Application Security Framework for DFS. *FIGI Security Clinic*, Geneva, Switzerland, December 2019.
 10. Security Research at the Florida Institute for Cybersecurity Research (with Patrick Traynor). *UF Eye Opener Breakfast*, Gainesville, FL, USA, October 2019.
 11. “Plug and Pray” Today: Reflecting on a Decade of Peripheral Security Research. *UCLA*, Los Angeles, CA, USA, April 2019.
 12. Cybersecurity Challenges in a Diverging Internet. *Jean Monnet Centre of Excellence Annual Workshop on the European Union*, Gainesville, FL, USA, April 2019.
 13. Designing a Security Management Framework for Digital Financial Services. *ITU Security, Infrastructure, Trust Working Group Meeting*, Washington, DC, USA, April 2019.
 14. Developing a Consortium on Trust in Media and Technology. *Invited Panelist. University of Florida Foundation National Board Meeting*, Atlanta, GA, USA, March 2019.
 15. Designing a Security Management Framework for Digital Financial Services. *Financial Inclusion Global Symposium*, Cairo, Egypt, January 2019.
 16. Staying Safe in Cyberspace: Understanding the Threats and Defenses. *The Village at Gainesville*, Gainesville, FL, USA, October 2018.
 17. Domain Informed Techniques for IoT Security (with Tuba Yavuz). *ARM Research*, San Jose, CA, USA, October 2018.
 18. A Discussion on Security Education in Academia. *Invited Panelist. ACM Conference on Communications and Communication Security (CCS)*, Toronto, ON, Canada, October 2018.
 19. New Attack Surfaces Against Mobile Platforms. *Columbia University Cybersecurity and Payments Roundtable*, New York, NY, August 2018.
 20. **Keynote Address:** Secure and Efficient Whole-System Data Provenance: A Retrospective. *2nd Workshop on Provenance-based Security*, London, UK, July 2018.
 21. “Plug and Pray” Today: Reflecting on a Decade of USB Security Research. *University of Oxford*,

Oxford, UK, July 2018.

22. “Plug and Pray” Today: Reflecting on a Decade of USB Security Research. *Chinese Academy of Sciences*, Beijing, China, June 2018.
23. Characterizing Tooling and Interfaces for Mobile Applications and Devices. *Samsung Research America*, Mountain View, CA, USA, May 2018.
24. “Plug and Pray” Today: Reflecting on 10 Years of USB Security Research. *Symantec Research Labs*, Culver City, CA, USA, May 2018.
25. A Security Framework for for DFS Featurephone Applications. *ITU FIGI Security, Information, and Trust Working Group Meeting*, Washington, DC, USA, April 2018.
26. Application Security for DFS Featurephone Applications. *Financial Inclusion Global Initiative Symposium*, Bangalore, India, November 2017.
27. Towards More Robust Interactions with Telephones and Peripherals (with Patrick Traynor). *Florida International University*, Miami, FL, October 2017.
28. **Keynote Address:** Making USB Safer for Hosts and Peripherals. *Third International Conference on Future Network Systems and Security (FNSS 2017)*, Gainesville, FL, August 2017.
29. Making USB Safer for Hosts and Peripherals. *ETH Zurich*, Zurich, Switzerland, July 2017.
30. Security Challenges in Mobile Devices. *FinTech, DFS and Payments Summit, Columbia Business School*, New York, NY, June 2017.
31. Webinar: Trends and Challenges in Mobile Money and Mobile Banking Application Security. *Digital Financial Services Observatory, Columbia University*, June 2017.
32. Towards Firmware Vetting on Legacy Embedded Platforms. *10th Workshop on Fault-Tolerant Spaceborne Computing Employing New Technologies*, Albuquerque, NM, May 2017.
33. Technical Deep Dive: Foundations of Hardware-Assisted Secure Multiparty Computation with SGX. *10th Workshop on Fault-Tolerant Spaceborne Computing Employing New Technologies*, Albuquerque, NM, May 2017.
34. Security and Privacy Challenges for Mobile Money Applications. *ITU Digital Financial Services Workshop: Exploring Innovation in Transactions and Financing in the Caribbean*, Port of Spain, Trinidad and Tobago, April 2017.
35. Everybody Jammin’ Up on Those Tings: Challenges for the Internet of Insecure Things. *Forum on Internet of Things: Smarter Living in the Caribbean*, Port of Spain, Trinidad and Tobago, April 2017.
36. Application Security for DFS Applications. *ITU Workshop on DFS and Financial Inclusion*, Washington, DC, April 2017.
37. **Keynote Address:** Making USB Safer for Hosts and Peripherals. *Computer Science Research Day, University of Georgia*, Athens, GA, April 2017.
38. Security in the DFS Ecosystem - Final Recommendations. *ITU Focus Group Meeting on Digital Financial Services*, Geneva, Switzerland, December 2016.
39. Making USB Great Again: New Defenses for Hosts and Devices. *MIT Lincoln Laboratory*, Lexington, MA, USA, October 2016.
40. Making USB Great Again: New Defenses for Hosts and Devices. *Dartmouth College*, Hanover, NH, USA, October 2016.

41. Recommendations for Securing Digital Finance. *ITU Focus Group Meeting on Digital Financial Services*, Dar es Salaam, Tanzania, September 2016.
42. Secure and Efficient Data Provenance. *DARPA Transparent Computing PI Meeting*, Cambridge, MA, USA, July 2016.
43. Security for Digital Alternative Data and SME Supply Chain Finance. *G20 Global Partnership for Financial Inclusion (GPFI) Forum*, Chengdu, China, July 2016.
44. Privacy-Preserving Computing Made Practical. *9th Workshop on Fault-Tolerant Spaceborne Computing Employing New Technologies*, Albuquerque, NM, USA, June 2016.
45. Host Fingerprinting for Identity and Integrity Assurance Using Machine Learning. *9th Workshop on Fault-Tolerant Spaceborne Computing Employing New Technologies*, Albuquerque, NM, USA, June 2016.
46. Security and Privacy Challenges for Mobile Money in Mobile Devices. *5th Columbia Business School Digital Financial Services and Emerging Payments Summit*, New York, NY, USA, May 2016.
47. Trustworthy Collection Architectures for Data Provenance. *Concordia University*, Montreal, QC, Canada, May 2016.
48. Analyzing the Security of Sticky SIMs. *ITU Focus Group Meeting on Digital Financial Services*, Washington, DC, USA, April 2016.
49. Secure and Trustworthy Computers, Devices, and Systems. *South Carolina Governor's School for Science and Mathematics*, Harstville, SC, USA, December 2015.
50. Security Recommendations for the DFS Ecosystem. *ITU Focus Group Meeting on Digital Financial Services*, Geneva, Switzerland, December 2015.
51. Securing Storage for Insider Threat Mitigation. *Florida Cyber Consortium Research Symposium*, Tampa, FL, USA, October 2015.
52. Tales From the Crypt(o): Spooky Stories of Ideal-World Cryptography Meeting Real-World Implementation. *University of Calgary*, Calgary, AB, Canada, November 2015.
53. Vulnerability Analysis of Branchless Banking Apps in the Developing World. *ITU Focus Group Meeting on Digital Financial Services*, Kuala Lumpur, Malaysia, October 2015.
54. New Trends in Cybersecurity: Securing Critical Cyber-Infrastructure. *US Department of State/Department of Justice Cybercrime and Cybersecurity Workshop for Lusophone Africa*, Maputo, Mozambique, September 2015.
55. Trustworthy Collection Architectures for Data Provenance. *MIT Lincoln Laboratory*, Lexington, MA, USA, August 2015.
56. Be Who You Are: Establishing Host Identity in Local and Remote Systems. *EURECOM Institute*, Sophia Antipolis, France, July 2015.
57. Be Who You Are: Establishing Host Identity in Local and Remote Systems. *Symantec Research Labs*, Mountain View, CA, USA, April 2015.
58. Be Who You Are: Challenges of Establishing Host Identity in Local and Remote Systems. *Rutgers University*, New Brunswick, NJ, USA, March 2015.
59. Certificate Validation and the Future of SSL Security. *Chi Sigma IO, University of Florida*, Gainesville, FL, USA, November 2014.

60. Forced Perspectives: Evaluating an SSL Trust Enhancement at Scale. *2014 ACM Internet Measurement Conference (IMC'14)*, Vancouver, BC, Canada, November 2014.
61. USB Host Fingerprinting. *Lockheed Martin*, Orlando, FL, October 2014.
62. Improving Security Through Collaborative Frameworks. *IEEE CollaborateCom (Invited Panelist)*, Miami, FL, October 2014.
63. Establishing Host Identity in Systems and Storage. *University of Florida*, Gainesville, FL, April 2014.
64. Establishing Host Identity Through Indirect Measurement. *Pennsylvania State University*, University Park, PA, April 2014.
65. **Keynote Address:** Trust No One? Addressing the Challenges of Bottom Up Trust in Systems and Storage. *Symantec Cutting Edge*, Springfield, OR, March 2014.
66. Leveraging Emerging Storage Functionality for New Security Services. *Oregon State University*, Corvallis, OR, June 2012.
67. Protecting Portable Storage with Host Validation. *Queen's University*, Kingston, ON, Canada, April 2012.
68. Protecting Portable Storage with Host Validation. *NSERC ISSNet Security Workshop*, Kingston, ON, Canada, April 2012.
69. Attacks and Defenses Against the Cloud Infrastructure. *Oregon Security Day*, Eugene, OR, April 2012.
70. Leveraging Emerging Storage Functionality for New Security Services. *Portland State University*, Portland, OR, November 2011.
71. Leveraging Emerging Storage Functionality for New Security Services. *Galois*, Portland, OR, August 2011.
72. Securing Homecare Rehabilitation Environments. *UW/MSR Research Retreat*, Cle Elum, WA, August 2011.
73. Challenges and Alternatives to the Host Identification Problem. *Oregon Security Day*, Eugene, OR, April 2011.
74. Porscha: Policy Oriented Secure Content Handling in Android. *ACSAC 2010*, Austin, TX, December 2010.
75. Kells: A Protection Framework for Portable Data. *ACSAC 2010*, Austin, TX, December 2010.
76. Protection Mechanisms for Portable Devices. *Symantec Research Labs*, Mountain View, CA, November 2010.
77. Challenges and Alternatives to the Host Identification Problem. *2nd ACM Northeast Forensics Exchange (NeFX2010)*, Washington, DC, September 2010.
78. Disk-Enabled Authenticated Encryption. *MSST 2010*, Incline Village, NV, May 2010.
79. Leveraging Emerging Storage Functionality for New Security Services. *Polytechnic Institute of New York University*, Brooklyn, NY, March 2010.
80. Leveraging Emerging Storage Functionality for New Security Services. *Simon Fraser University*, Surrey, BC, Canada, March 2010.

81. Leveraging Emerging Storage Functionality for New Security Services. *Stevens Institute of Technology*, Hoboken, NJ, March 2010.
82. Leveraging Emerging Storage Functionality for New Security Services. *University of Tennessee*, Knoxville, TN, March 2010.
83. Leveraging Emerging Storage Functionality for New Security Services. *University of Oregon*, Eugene, OR, March 2010.
84. Leveraging Emerging Storage Functionality for New Security Services. *Naval Postgraduate School*, Monterey, CA, February 2010.
85. Towards a Secure and Efficient System for End-to-End Provenance. *TaPP'10*, San Jose, CA, February 2010.
86. Rootkit-Resistant Disks. *Security Reading Group*, *University of Toronto*, Toronto, ON, Canada, November 2009.
87. Rootkit-Resistant Disks. *Digital Security Seminar*, *Carleton University*, Ottawa, ON, Canada, November 2009.
88. U Can't Touch This: Protections for Portable Storage. *IWSSPS 2009*, Grenoble, France, Oct. 2009.
89. FlowStor: Preserving Information Flow on Portable Storage Devices. *Symantec Corporation*, Mountain View, CA, USA, August 2009.
90. Leveraging Emerging Disk Functionality for New Security Services. *Symantec Research Labs*, Mountain View, CA, USA, June 2009.
91. Rootkit-Resistant Disks. *CCS'08*, Alexandria, VA, USA, October 2008.
92. Experimenting with BGP Security Mechanisms Using the DETER Framework. *Boeing Integrated Defense*, Philadelphia, PA, USA, October 2008.
93. Terabyte Home Security and Access Control. *Seagate Research*, Pittsburgh, PA, USA, August 2008.
94. Towards Automated Privilege Separation. *ICISS 2007*, Delhi, India, December 2007.
95. Non-Volatile Memory and Disks: Avenues for Policy Architectures. *CSAW 2007*, Fairfax, VA, USA, November 2007.
96. Autonomously Secure Disks. *Software Reading Group*, *University of Michigan*, Ann Arbor, MI, USA, September 2007.
97. Leveraging Non-Volatile Memory for Storage Security. *2007 USENIX Security Symposium, Works in Progress*, Boston, MA, USA, August 2007.
98. Analysis of the IPv4 Address Delegation Structure. *ISCC 2007*, Aveiro, Portugal, July 2007.
99. Performance Optimizations of IPsec for Storage. *2007 SNIA Summer Security Summit*, Pittsburgh, PA, June 2007.
100. Optimizing BGP Security by Exploiting Path Stability. *CCS'06*, Alexandria, VA, November 2006.
101. Threats and Attacks in Interdomain Routing. *P2INGS Quarterly Meeting*, Tempe, AZ, February 2004.
102. Network Operations in Tier-1 Internet Service Providers. *Columbia University*, New York, NY, March 2001.
103. Network Operations in Tier-1 Internet Service Providers. *Telcordia Technologies*, Morristown, NJ,

August 2000.

Press Coverage

Research-Related Coverage

1. “Exploiting Decades-Old Telephone Tech to Break Into Android Devices.” Lily Hay Newman, *Wired*, August 29, 2018. <https://www.wired.com/story/at-commands-android-vulnerability/>
2. “Modern smartphones vulnerable to old-school attack.” Francis Navarro, *Komando.com*, August 28, 2018. <https://www.komando.com/happening-now/483269/modern-smartphones-vulnerable-to-old-school-attack>
3. “ATak na Androida: modemowe komendy standardu z lat 80 pozwalają przejąć kontrolę nad smartfonem” (Translation from Polish: “ATtack on Android: modem commands from the 80’s standard allow you to take control of your smartphone”.) Adam Golanski, *Dobreprogramy*, August 28, 2018. <https://www.dobreprogramy.pl/ATak-na-Androida-modemowe-komendy-standardu-z-lat-80-pozwalaja-przejac-kontrolę-nad-smartfonem,News,90526.html>
4. “Смартфоны ASUS, HTC, Huawei, Lenovo, LG, Samsung, Sony легко взломать технологией, созданной для древнего модема Подробнее” (Translation from Russian: “Smartphones ASUS, HTC, Huawei, Lenovo, LG, Samsung, Sony can easily crack technology created for the ancient modem.”) Roman Georgiev, *CNews Russia*, August 28, 2018. http://safe.cnews.ru/news/top/2018-08-28_smartfony_asushtchuaweilenovolsamsungsony
5. “AT Command Hitch Leaves Android Phones Open to Attack.” Lindsey O’Donnell, *Threatpost*, August 27, 2018. <https://threatpost.com/at-command-hitch-leaves-android-phones-open-to-attack/136938/>
6. “New security risk for smartphones bring you a ‘ghost user’.” Ida Torres, *Android Community*, August 27, 2018. <https://androidcommunity.com/new-security-risk-for-smartphones-brings-you-a-ghost-user-20180827/>
7. “Android at the mercy of AT commands.” Nick Farrell, *Fudzilla*, August 27, 2018. <https://fudzilla.com/news/mobile/47037-android-at-the-mercy-of-at-commands>
8. “Android-Smartphones durch Modem-Befehle verwundbar” (Translation from German: “Android Smartphones vulnerable through modem commands”.) Sebastian Gruner, *Golem.de: IT-News Für Profis*, August 27, 2018. <https://www.golem.de/news/at-commands-android-smartphones-durch-modem-befehle-verwundbar-1808-136205.html>
9. “Telefony s Androidem lze ovládnout přes USB pomocí AT příkazů”(Translation from Czech: “Android phones can be controlled via USB using AT commands”). Petr Krcmar, *Root.cz Linux News*, August 27, 2018. <https://www.root.cz/clanky/telefony-s-androidem-lze-ovladnout-pres-usb-pomoci-at-prikazu/>
10. “У 11 производителей Android-смартфонов обнаружили уязвимость к АТ-командам” (Translation from Russian: “11 manufacturers of Android-smartphones found a vulnerability to AT-commands”.) *Tproger*, August 27, 2018. <https://tproger.ru/news/at-commands-deface-smartphones/>
11. “Smartphones From 11 OEMs Vulnerable to Attacks via Hidden AT Commands.” Catalin Cimpanu,

Bleeping Computer, August 25, 2018. <https://www.bleepingcomputer.com/news/security/smartphones-from-11-oems-vulnerable-to-attacks-via-hidden-at-commands/>

12. "Smartphone security risk compared to 'having a ghost user on your phone'." Jonathan Griffin, *University of Florida News*, August 22, 2018. <http://news.ufl.edu/articles/2018/08/smartphone-security-risk-compared-to-having-a-ghost-user-on-your-phone.php>
13. "Ransomware 'stopped' by new software." *BBC News*, July 2016. <http://www.bbc.com/news/technology-36772461>
14. "United States and Mozambique Host Cybersecurity and Cybercrime Workshop in Maputo for Lusophone Africa." *US Department of State*, September 2015. <http://www.state.gov/r/pa/prs/ps/2015/09/247088.htm>
15. "Researchers Find Security Flaws in Developing-World Money Apps." Jennifer Valentino-Devries. *Wall Street Journal*, August 2015. <http://blogs.wsj.com/digits/2015/08/11/researchers-find-security-flaws-in-developing-world-money-apps/>
16. "Hack could let browsers use cloud to carry out big attacks on the cheap." Dan Goodin, *Ars Technica*, November 2012. <http://arstechnica.com/security/2012/11/hack-could-let-browsers-use-cloud-to-carry-out-big-attacks-on-the-cheap/>

Other Coverage

17. "Voters in Two States Report Threatening 'Vote for Trump' Emails", Nick Corasanitti, Ben Decker, and Stephanie Saul, *The New York Times*, 20 October 2020. <https://www.nytimes.com/2020/10/20/us/politics/florida-alaska-trump-emails.html>
18. "What Would It Take to Shut Down the Entire Internet?" Daniel Kolitz, *Gizmodo*, 30 September 2019. <https://gizmodo.com/what-would-it-take-to-shut-down-the-entire-internet-1837984019>
19. "Devious Ransomware Frees You If You Infect Two Other People." Lily Hay Newman, *Wired*, 13 December 2016. <https://www.wired.com/2016/12/popcorn-time-ransomware/>
20. "UF experts: iPhone issue could set precedent." Christopher Curry and Dahlia Ghabour, *Gainesville Sun*, 26 February 2016. <http://www.gainesville.com/article/20160226/ARTICLES/160229743/-1/archive?Title=UF-experts-iPhone-issue-could-set-precedent>
21. "Biometric Security Comes To Your Smartphone." Mitch Kroner, *Think Out Loud, Oregon Public Broadcasting*, 24 September 2013. <http://www.opb.org/radio/programs/thinkoutloud/segment/biometric-security-comes-to-your-smartphone/>

Previous Academic Appointments

Associate Professor, Department of Computer and Information Science and Engineering, University of Florida, Aug. 2014 - Aug. 2021

Associate Director of the Florida Institute for Cybersecurity Research. Previous co-director of the Southeastern Security for Enterprise and Infrastructure Center.

Assistant Professor, Department of Computer and Information Science, University of Oregon, Sep. 2010 - Aug. 2014

Research and teaching classes in systems security. Founder and leader of the Oregon Systems

Infrastructure Research and Information Security Laboratory. Co-director of the Center for Cyber Security.

- **Leadership:** Organizer of Computer Security Day at the University of Oregon (2014, 2013, 2012, 2011, recipient of CAS program grant for three years), liaison to Oregon Engineering and Technology Council (2010-2012), faculty advisor to UO Security Club (2012-2014), faculty liaison to Women in Computer Science (2012-2013).
- **Committee Service:** University Committee on General Science Major (2014), Graduate Education Committee (2010-2012, 2014-present), Computing Resources Committee (2011-2012), Industrial Affairs Committee (2012-2014), Strategic Planning Committee (2013).

Research Assistant, Department of Computer Science, Pennsylvania State University, Jan. 2005 - Aug. 2010

- **Senior Personnel**, NSF HECURA: Collaborative Research: Secure Provenance in High-End Computing Systems, NSF (CCF), \$307,073, Aug. 2009 - present.
- Funded by NSF HECURA grant 0621429 from Aug. 2006 - Jul. 2010, investigating security of storage systems.
- Funded by NSF grant CNS-0335241 from Aug. 2005 - Aug. 2006, investigating BGP security solutions and large-scale simulation.

Lead Graduate Student, Systems and Internet Infrastructure Security (SIIS) Laboratory, Pennsylvania State University, Sep. 2007 - Dec. 2008

- Administrative and logistical lead student in the SIIS lab. Ran weekly lab meetings of 13 members, met with students individually for mentoring and development of leadership and research skills. Responsible for most daily operational activities in the lab.

Instructor, Department of Electrical Engineering, Pennsylvania State University, Jun.-Aug. 2006

- Instructor for *EE/CSE 458*, the senior undergraduate course on computer networking.

Guest Lecturer, Department of Computer Science, Pennsylvania State University, 2005-present

- *CSE 543 (Computer Security)*: presented lectures on cryptography and public key infrastructures.
- *CSE 597A (Advanced Topics in System Security)*: lectured on virtual machine security.
- *CSE 411 (Operating Systems)*: presented lectures on storage systems.
- *CSE 545 (Advanced Network Security)*: lectured on BGP security.

Teaching Assistant, Stern School of Business, New York University, Fall 2005

- Lectured, graded, and provided logistics for Computer and Network Security course offered for MBA students.

Teaching Assistant, Department of Computer Science and Engineering, Pennsylvania State University, Fall 2004

- Held lab hours and graded for Computer System Architecture course.

Graduate Research Assistant, Department of Computer Science, Columbia University, Oct. 2001 - May 2002

- Worked on SIP security, implementing authentication into the Panasonic PINTL JAIN SIP stack.

Teaching Assistant, Department of Computer Science, Queen's University, Fall 1998

- Held office hours and graded for Introduction to Computer Science course.

Industrial Experience

Co-Founder and Chief Operating Officer, CryptoDrop LLC, 2017-2019.

- Anti-ransomware startup based in Gainesville, Florida.

Research Intern, Symantec Research Labs, Mountain View, CA, Jun - Aug. 2009

- Examined host-based validation for portable storage devices, and implemented a prototype using embedded Linux.

Research Intern, Seagate Research, Pittsburgh, PA, May - Aug. 2008

- Developed security infrastructures for distributed storage in a home user environment, focussing on distributed access control mechanisms.

Research Intern, AT&T Labs - Research, Florham Park, NJ, Dec. 2003 - Aug. 2004

- Investigated security threats in BGP and created an overview of threats and currently deployed solutions. Developed cryptographic constructions allowing efficient security solutions and performed simulations based on trace data.

Technical Intern, Flarion Technologies, Bedminster, NJ, Summer 2002

- Implemented packet diversion and IP header modification program in BSD for mobile IP packet testing.

Research Scientist, Telcordia Technologies, Morristown, NJ, Sep. 2000 - Nov. 2001

- Built and maintained an HFC lab to emulate a cable plant; researched routing and service assurance methodologies for multiple service providers accessing common transport.

Technical Specialist, UUNET, Toronto, ON, May 1999 - Aug. 2000

- Monitored and troubleshot intermediate and backbone connections across the UUNET Canadian network, as well as dealing with network security and web server issues.

Summer Intern, Royal Bank of Canada, Toronto, ON, Summer 1997, 1998

- Investigated financing opportunities for life science and high tech companies through financial and technical analysis; ported applications from mainframe to client-server environments.