UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,

Petitioner,

v.

HEADWATER RESEARCH LLC,

Patent Owner.

Patent No. 9,647,918
Case No. IPR2026-00137

DECLARATION OF DR. HENRY HOUH

Table of Contents

I. INTRODUCTION 1

II. SUMMARY OF OPINIONS 1

III. QUALIFICATIONS AND BACKGROUND 2

IV. MATERIALS CONSIDERED 10

V. LEGAL STANDARDS 12

A. Claim Construction 13

B. Anticipation Under 35 U.S.C. § 102 14

C. Obviousness Under 35 U.S.C. § 103 14

D. Secondary Considerations of Non-Obviousness 17

VI. TECHNOLOGY OVERVIEW 19

VII. THE '918 PATENT 21

A. Specification 21

B. Prosecution History 24

VIII. CLAIM CONSTRUCTION 25

IX. LEVEL OF ORDINARY SKILL 26

X. UNPATENTABILITY OF THE CHALLENGED CLAIMS 27

A. Ground 1: *Bennett* and *Vadde* render Claims 1-3, 8-9, 13-14, and 19 Obvious 27

1. Overview of *Bennett* 27

2. Overview of *Vadde* 28

3. Motivation to Combine *Bennett* and *Vadde* with Reasonable Expectation of Success 29

4. Claim 1 36

5. Claim 2: "*The wireless end-user device of claim 1, wherein to associate wireless network data usage for the media object network data transfers with the device application that makes the data transfer request for the media object comprises to identify at least one of an application name, an application identifier, or a process identifier for the application that makes the data transfer request."*.....75

6. Claim 3: *"The wireless end-user device of claim 2, wherein the data transfer request comprises a network resource identifier that identifies a source of the data to be transferred, a proxy to the source of the data to be transferred, or the media object to be transferred, in particular, wherein the network resource identifier comprises one or more of an Internet Protocol address, a Uniform Resource Locator, a remote file name/address, a stream name, and an object name."* 76

7. Claim 8: *"The wireless end-user device of claim 3, wherein to manage network data transfers for the media object by interfacing with the at least one network stack comprises to map the data transfer request and network resource identifier to one or more data flow connections communicated through the device network stack."*.....78

8. Claim 9: *"The wireless end-user device of claim 1, further comprising a media player and a user interface, wherein the media object comprises media data that is, as a result of the media service manager management of network data transfers for the media object, received by the device and played by the media player through the user interface."* 79

9. Claim 13: *"The wireless end-user device of claim 1, the one or more service classification and measurement agents to further associate one or more traffic flows, comprising the media object network data transfers, with the device application that makes the data transfer request, the device further comprising an enforcement agent to, based on the association between the one or*

more traffic flows and the device application, enforce an application-based usage control on network data usage by one or more of the device applications." 83

10. Claims 14 and 19: 85

B. Ground 2: Claims 4-6, 11, and 15-17 Are Obvious over *Bennett* in View of *Vadde* and *Riggs* 92

1. Overview of *Riggs* 92

2. Motivation to Combine *Bennett-Vadde* and *Riggs* with Reasonable Expectation of Success 93

3. Claim 4: *"The wireless end-user device of claim 3, wherein to associate wireless network data usage for the media object network data transfers with the device application that makes the data transfer request for the media object further comprises to store an entry comprising the at least one of the application name, the application identifier, or the process identifier for each of the device applications that makes a data transfer request, each stored entry further comprising information about the corresponding network resource identifier for the data transfer request."* 97

4. Claim 5: *"The wireless end-user device of claim 4, wherein the one or more service classification and measurement agents includes a requesting application storing function within the media services manager."* 99

5. Claim 6: *"The wireless end-user device of claim 5, wherein the media service manager is a first media service manager and the requesting application storing function is a first requesting application storing function, the device further comprising a second media service manager of a different type than the first media service manager, the service classification agent including a second requesting application storing function within the second media service manager."* 101

6. Claim 11: 104

7. Claims 15-17: 110

C. Ground 3: Claims 7, 12, and 18 Are Obvious over *Bennett* in View of *Vadde*, *Riggs*, and *Hendrickson* 115

1. Overview of Hendrickson 115

2. Motivation to Combine *Bennett-Vadde-Riggs* and *Hendrickson* with Reasonable Expectation of Success 116

3. Claim 7: *"The wireless end-user device of claim 6, further comprising a usage and classification database, the one or more service classification and measurement agents to receive application association information stored by the first and second requesting application storing functions, and to maintain the usage and classification database based in part on the received application association information."* 119

4. Claim 12: 122

5. Claim 18: *"The wireless end-user device of claim 17, further comprising a usage and classification reconciliation agent and usage and classification database, the usage and classification reconciliation agent to receive application association information stored by the first and second requesting application storing functions, and to maintain the usage and classification database based in part on the received application association information."* 126

D. Ground 4: Claim 10 Is Obvious over *Bennett* in View of *Vadde*, *Riggs*, and *Srikantan* 128

1. Overview of *Srikantan* 128

2. Motivation to Combine *Bennett-Vadde-Riggs* and *Srikantan* with Reasonable Expectation of Success 129

3. Claim 10: 131

XI. CONCLUSION 137

I. INTRODUCTION

1. I have been retained on behalf of Google LLC ("Petitioner") to offer technical opinions related to U.S. Patent No 9,647,918 ("the '918 patent," EX1001). I understand that Petitioner is requesting that the Patent Trial and Appeal Board ("PTAB" or "Board") institute an inter partes review ("IPR") proceeding of the '918 patent.

2. I have been asked to provide my independent analysis of the '918 patent based on the prior art publications cited in this declaration.

3. I am not and have never been an employee of Petitioner. I received no compensation for this declaration beyond my normal hourly compensation based on my time actually spent analyzing the '918 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 '918 patent.

II. SUMMARY OF OPINIONS

4. This declaration considers claims 1-19 ("Challenged Claims") of the '918 patent. Below, I set forth the opinions I have formed, the conclusions I have reached, and the bases for these opinions and conclusions. I believe the statements contained in this declaration to be true and correct to the best of my knowledge.

5. Based on my experience and knowledge of the art at the time of the applicable priority date, it is my opinion that the Challenged Claims would have been obvious to one of ordinary skill in the art based on the grounds below:

Ground	Claims Challenged	Basis	Reference(s)
1	1-3, 8-9, 13-14, and 19	§103	*Bennett* in view of *Vadde*
2	4-6, 11, and 15-17	§103	*Bennett* in view of *Vadde* and *Riggs*
3	7, 12 and 18	§103	*Bennett* in view of *Vadde*, *Riggs* and *Hendrickson*
4	10	§103	*Bennett* in view of *Vadde, Riggs* and *Srikantan*

6. I understand that *Bennett* (EX1041), *Vadde* (EX1042), *Riggs* (EX1043), *Hendrickson* (EX1054), and Srikantan (EX1055) are prior art to the '918 patent.

III. QUALIFICATIONS AND BACKGROUND

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

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

9. As indicated on my Curriculum Vitae, which I understand is being submitted as EX1004, I received a Ph.D. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology ("MIT") in 1998. I also received a Master of Science degree in Electrical Engineering and Computer Science in 1991, a Bachelor of Science Degree in Electrical Engineering and Computer Science in 1989, and a Bachelor of Science Degree in Physics in 1990, all from MIT.

10. During my college studies, I focused on communications and data networking. I took specialized courses including graduate courses in telecommunications networks, optical communications, data networking, media encoding, and interactive video. I, along with other graduate students in a networking research group, maintained both the computer workstations and the networking devices in the research group.

11. I also served as a teaching assistant for the Computation Structures core course a number of times. At the time, the course involved teaching computer architecture at a low level, and the laboratory component involved building a CPU from discrete components at the gate level and higher. When I was enrolled in this course as an undergraduate student, I redesigned the hardware and software, including microcode and nanocode of this CPU, to achieve an average speed up of roughly thirty times. Based in part on this, I became a laboratory assistant, head laboratory assistant, and then a teaching assistant in my second term senior year,

which is unusual in that teaching assistantships at MIT are normally reserved for graduate students. I further became head teaching assistant and also taught some recitation sections.

12. As part of my doctoral research at MIT from 1991-1998, I worked as a research assistant in the Telemedia Network Systems ("TNS") group at the Laboratory for Computer Science. The TNS group built a high-speed gigabit ATM network and applications which ran over the network, such as remote video capture (including audio), processing and display on computer terminals. In addition to helping design the core network components (such as the ATM switch), designing and building the high-speed ATM links, and designing and writing the device drivers for the interface cards, I also set up the group's web server, which at the time was one of the first several hundred web servers in existence. I used programmable logic devices, such as programmable array logic and field programmable gate arrays, and their software development environments, in the design of these components. Our high-speed data network carried multimedia data including video and audio data within ATM cells.

13. The TNS group was the first group to initiate a remote video display over the World Wide Web. Vice President Al Gore visited our group in 1996 and received a demonstration of—and remotely drove—a radio-controlled toy car with a wireless video camera mounted on it; the video was encoded by TNS-designed

hardware, streamed over the TNS-designed network and displayed using TNS-designed software.

14. I authored or co-authored twelve papers and conference presentations on our group's research. I also co-edited the final report of the gigabit networking research effort with Professor David Tennenhouse and Senior Research Scientist David Clark. David Clark is generally considered to be one of the fathers of the Internet Protocol and served as Chief Protocol Architect for the Internet. With its focus on networking, the group, including myself, set up and maintained the network and computer systems. These systems included the networking on the workstations and desktops, the distributed file system, desktops and workstations, setting up and maintaining the distributed file system (Network File System) and the authentication system (Network Information Service, formerly known as Yellow Pages).

15. I defended and submitted my Ph.D. thesis, titled "'Designing Networks for Tomorrow's Traffic," in January 1998. As part of my thesis research, I analyzed local-area and wide-area traffic flows to show a more efficient method for routing packets in a network, based on traffic patterns at the time. My thesis also addressed real-time streamed audio and video. The network traffic that I analyzed was IP protocol traffic, including UDP and TCP.

16. I have been involved in Voice over Internet Protocol ("VoIP") technologies since 1997 and have specific experience in designing and testing VoIP networks.

17. From 1997 to 1999, I worked at NBX Corporation, which was acquired by 3Com Corporation in 1999. During this time, I was a Senior Scientist and Engineer working in IP Telephony. NBX delivered the world's first folly featured business telephone system to run over a data network, the NBX100. NBX was one of the first business phone systems to be configurable via a web interface. Users and administrators had access to varying levels of configuration for the phone system.

18. As part of my work at NBX, I designed the core audio reconstruction algorithms for the telephones which depacketized the voice data and reconstructed the audio. ln addition, I designed the voice data packet transmission algorithms. I created a system to capture and analyze network packets and voice traffic flows sent by devices in the NBX system for aid in testing and debugging. I also designed and validated the core packet transport protocol used by the phone system. In addition, I designed and oversaw the development of the underlying transport protocol used by the NBXl00 phone system for reliable packet transport. That transport protocol is still used by the NBX100 system and its successors and is estimated to be used hundreds of millions of times daily. I wrote NBX's first demonstration IP software stack, which added the capability for utilizing the NBX100 phone system on an IP

network. NBX first demonstrated a phone in the NBX100 system working over the Internet in 1998 at a trade show in Las Vegas. I was later the lead architect in designing NBX's next-generation highly scalable system, and, after NBX was acquired by 3Com, I did some work with 3Com's cable equipment division, including demonstrating a working NBX IP phone system over 3Com's cable equipment infrastructure using an early version of DOCSIS at a trade show in 1999. The NBX100 was the market's leading business phone system to run on a data network for several years following its introduction. During that time, I became more familiar with the various standards relevant to Internet telephony as well as the problems which designers of commercial telephony operations were faced with in implementing VoIP.

19. I, along with two of NBX's founders, obtained U.S. Patent No. 6,697,963 titled "Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data," for some of the work we did while at NBX

20. After NBX, I worked at Teradyne, a test tool company primarily focused on semiconductors. Teradyne had recently acquired Hammer, a company that specialized in load and functional testing for telecommunications systems. The Hammer product is well known as a telecom test tool. Teradyne spun out Hammer and several other internal divisions into an independent company called Empirix. I

became Chief Technologist of the Hammer division of Empirix. Empirix was a leader in VoIP network testing and monitoring.

21. At Empirix, I laid out a new multi-year product vision for data network testing, secured internal funding for the effort, and led a team to deliver a new technology platform to the market in February 2001. This new product, Packet Sphere, initially emulated network behavior so that wide-area VoIP connections could be tested in a lab. A later release allowed PacketSphere to generate high volumes of VoIP calls, including media streams, and to monitor the quality of VoIP voice streams. Later, the core technology, was added to other Empirix products such as Empirix's Hammer XMS to monitor thousands of VoIP media streams in real time to determine their quality. PacketSphere was Empirix's most successful new platform introduction. Companies purchased the PacketSphere product to emulate an Internet Protocol network to see the effects of deploying their product on the Internet prior to launch. PacketSphere received several industry awards.

22. During my time at Empirix, I presented lectures on VoIP and data network testing to companies including Lucent Labs (formerly AT&T Bell Labs). I was also invited to present several guest lectures in a software engineering course at MIT. Since then, I have also participated twice as a unit lecturer (two weeks) in an experimental course that was taught by an Institute Professor (the highest award that

a: MIT Professor can achieve) and sponsored by the Chairman of the MIT Corporation (MIT's board of trustees).

23. From 2004 to 2008, I was employed by BBN Technologies Corp., a technology research and development company located in Cambridge, Massachusetts. BBN Technologies is a world-renowned company with expertise in acoustics, speech recognition, and communications technology. BBN Technologies staff have pioneered many internetworking technologies and Internet applications, and built some of the world's largest government and commercial data networks.

24. My duties and responsibilities at BBN Technologies generally included commercialization of the technologies developed by BBN Technologies, which included spinning off companies and growing commercial businesses in-house. More particularly, I was involved in utilizing the award-winning AVOKE STX speech recognition technology to create the public audio/video search engine EveryZing (formerly known as PodZinger) which was spun out into a stand-alone company now known as RAMP, Inc. PodZinger won the 2006 MITX Technology Award for best Web 2.0 Application and was also named the 2006 Forbes Favorite Video & Audio Search Engine, beating out Google, Yahoo, and other companies. After managing the creation of the initial prototype system, PodZinger built out a full streaming audio and video search solution when I was the Vice President of Operations and Technology there. Our solution also included providing video

hosting and streaming services. I was also involved in the Boomerang Mobile Shooter Detection project as the Vice President of Engineering for the program. The Boomerang system was deployed to Iraq and Afghanistan, and was credited with saving many lives.

25. From 1989 to 1990, I worked at AT&T Bell Laboratories on optical computers. This work generated six peer-reviewed papers, and multiple U.S. and European patent applications in which I was named as a co-author or inventor. I also interned at AT&T Bell Laboratories in 1987 and 1988. Additional relevant experience is listed in my Curriculum Vitae.

26. I am a named inventor on several patents and published patent applications that are related to streaming technology including: U.S. Patent No. 6,967,963, entitled "Telecommunication Method for Ensuring On-time Delivery of Packets Containing Time- Sensitive Data" and U.S. Patent Application Publication No. 2002/0015387, entitled "Voice Traffic Packet Capture and Analysis Tool for a Data Network."

IV. MATERIALS CONSIDERED

27. In forming my opinions, I have reviewed the following documents and any other document cited in this declaration.

Exhibit	Description
EX1001	U.S. Patent No. 9,647,918 to Gregory G. Raleigh, et al. ("the '918 patent").
EX1002	Excerpts from the Prosecution History of the '918 patent ("the Prosecution History").
EX1005	Infringement Contentions, *Headwater Research LLC v. Samsung Electronics Co., Ltd. et al.,* 2-23-cv-00641 (EDTX), filed May 15, 2024.
EX1013	U.S. Provisional Application No. 61/381,162
EX1041	U.S. Patent Publication No 2006/0149811 ("Bennett")
EX1042	U.S. Patent Publication No. 2012/0117478 ("Vadde")
EX1043	U.S. Patent No. 8,429,516 ("Riggs")
EX1044	EP Patent Publication No. 1 850 575 A1 ("Rybak")
EX1045	U.S. Patent Publication No. 2004/0260630 ("Benco")
EX1046	U.S. Patent No. 6,578,077 ("Rakoshitz")
EX1047	U.S. Patent Publication No. 2006/0223495 ("Cassett")
EX1048	U.S. Patent Publication No. 2008/0080458 ("Cole")
EX1049	U.S. Patent Publication No. 2008/0209451 ("Michels")
EX1050	U.S. Patent Publication No. 2006/0039354 ("Rao")
EX1053	U.S. Provisional Application No. 61/435,564
EX1054	U.S. Patent No. 6,754,470 ("Hendrickson")
EX1055	U.S. Patent Publication No. 2002/0056126 ("Srikantan")
EX1056	Newton's Telecom Dictionary, 24th Edition
EX1057	Webster's New World, Telecom Dictionary

Exhibit	Description
EX1058	Wiley Electrical and Electronics Engineering (IEEE) Dictionary
EX1059	The Authoritative Dictionary of IEEE Standards Terms
EX1060	Claim Construction Order, *Headwater Rsch. LLC v. Samsung Elecs. Co. Ltd.*, 2:23-cv-00641-JRG-RSP (E.D. Tex. Apr. 7, 2025).
EX1061	J. Postel, Transmission Control Protocol, RFC 793, Internet Engineering Task Force (Sept. 1981) ("RFC 793")
EX1062	U.S. Patent Publication No. 2008/0122796 ("Jobs")
EX1063	U.S. Patent Publication No. 2010/0017506 ("Fadell")
EX1064	R. Braden, Requirements for Internet Hosts – Communication Layers, RFC 1122, Internet Engineering Task Force (Oct. 1989) ("RFC 1122")
EX1065	T. Berners-Lee et al., Uniform Resource Locators (URL), RFC 1738, Internet Engineering Task Force (Nov. 1995) ("RFC 1738")
EX1066	M. Handley et al, SIP: Session Initiation Protocol, RFC 2543, The Internet Society (Mar. 1999) ("RFC 2543")
EX1067	T. Berners-Lee & D. Connolly, Hypertext Markup Language - 2.0, RFC 1866, Internet Engineering Task Force (Nov. 1995) ("RFC 1866")

V. LEGAL STANDARDS

28. In forming my opinions and considering the subject matter of the '918 patent and its claims in light of the prior art, I am relying on certain legal principles that counsel in this case explained to me. My understanding of these concepts is summarized below.

29. I understand that the claims define the invention. I also understand that an unpatentability analysis is a two-step process. First, the claims of the patent are construed to determine their meaning and scope. Second, after the claims are construed, the content of the prior art is compared to the construed claims.

30. I understand that there are two ways in which prior art may render a patent claim unpatentable. First, I have been informed that the prior art can "anticipate" a claim. Second, I have been informed that the prior art can render a claim "obvious" to a person of ordinary skill in the art. I understand that a claim is patentable if it was not anticipated and would not have been obvious by the prior art at the effective filing date of the patent.

A. Claim Construction

31. I understand that the United States Patent and Trademark Office interprets claim terms in an *inter partes* review proceeding under the same claim construction standard that is used in a United States federal court. I understand that under this standard, the meaning of claim terms is considered from the viewpoint of one of ordinary skill in the art at the time of the alleged invention.

32. I have been informed that claim terms are generally given their ordinary and customary meaning as understood by one of ordinary skill in the art in light of the specification and the prosecution history pertaining to the patent. I understand,

however, that claim terms are generally not limited by the embodiments described in the specification.

33. I understand that in addition to the claims, specification, and prosecution history, other evidence may be considered to ascertain the meaning of claim terms, including textbooks, encyclopedias, articles, and dictionaries. I have been informed that this other evidence is often less significant and less reliable than the claims, specification, and prosecution history.

B. Anticipation Under 35 U.S.C. § 102

34. I understand that a patent claim is invalid as anticipated if a single piece of prior art teaches every element of the claims, viewed from the perspective of a person of ordinary skill in the art. I also understand that an anticipatory reference does not have to recite word for word what is in the anticipated claims. Anticipation can also occur when a claimed limitation is inherent in the relevant reference. I have been advised that if the prior art necessarily functions in accordance with, or includes, the claims limitations, it can anticipate even though the limitation is not expressly disclosed.

C. Obviousness Under 35 U.S.C. § 103

35. I understand that a patent claim is invalid as obvious if the claimed invention would have been obvious to a person of ordinary skill in the art at the time the claimed invention was made. This means that even if all of the elements of the

claim cannot be found in a single prior art reference that would anticipate the claim, a person of ordinary skill in the field who knew about all the prior art would have come up with the claimed invention. I understand that in an obviousness determination, the person of ordinary skill in the art is presumed to have knowledge of all prior art. I understand that whether a claim is obvious is based upon the determination of several factual issues.

36. I have been informed that, in an obviousness analysis, prior art must be analogous to the patent being considered. I have been informed that a prior art reference is considered to be analogous, or in the same field of art, if the reference is either (1) in the same field of endeavor as the challenged patent, regardless of the problems the challenged patent and the prior art address, or (2) reasonably pertinent to the particular problem being solved by the challenged patent.

37. In considering obviousness, I understand that one must determine the scope and content of the prior art. I also understand that one must determine what differences, if any, existed between the claimed invention and the prior art.

38. I understand that a patent claim composed of several elements is not proved obvious merely by demonstrating that each of its elements was independently known in the prior art. In evaluating whether such a claim would have been obvious, one may consider whether a reason has been identified that would have prompted a person of ordinary skill in the art to combine the elements or concepts from the prior

art in the same way as in the claimed invention. There is no single way to define the line between true inventiveness on the one hand (which is patentable) and the application of common sense and ordinary skill to solve a problem on the other hand (which is not patentable). For example, market forces or other design incentives may be what produced a change, rather than true inventiveness.

39. I also understand that, while there is no requirement that the prior art contain an express suggestion to combine known elements to achieve the claimed invention, and while a suggestion to combine known elements to achieve the claimed invention may come from the prior art as a whole or individually and may consider the inferences and creative steps a person of ordinary skill in the art would employ, as filtered through the knowledge of one skilled in the art, obviousness grounds cannot be sustained by mere conclusory statements and must include some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. Moreover, I understand that a proper obviousness analysis requires a determination of whether a person of ordinary skill in the art would have a "reasonable expectation of success"—but not "absolute predictability" of success—in achieving the claimed invention by combining prior art references.

40. I understand that one may consider whether (1) the change was merely the predictable result of using prior art elements according to their known functions, or whether it was the result of true inventiveness; (2) there is some teaching or

suggestion in the prior art to make the modification or combination of elements claimed in the patent; (3) the innovation applies a known technique that had been used to improve a similar device or method in a similar way; (4) the claimed invention would have been obvious to try, meaning that the claimed innovation was one of a relatively small number of possible approaches to the problem with a reasonable expectation of success by those skilled in the art; (5) the invention merely substituted one known element for another known element in order to obtain predictable results; (6) the invention merely applies a known technique to a known device, method, or product to yield predictable results, or (7) known work in the field may have prompted variations of use of the same inventions in the same or different fields due to market forces or design incentives that would have been predictable to a person of ordinary skill in the art.

41. I understand that one must be careful not to determine obviousness using the benefit of hindsight; many true inventions might seem obvious after the fact. I understand that the determination should be based on the position of a person of ordinary skill in the field at the time the claimed invention was made and should not consider what is known today or what is learned from the teaching of the patent.

D. Secondary Considerations of Non-Obviousness

42. I understand that certain secondary considerations may be considered in evaluating obviousness in order to prevent hindsight bias. These secondary

considerations include commercial success of products that practice the patent, long-felt need for the patented technology, failure by others to solve the problem addressed by the patent, initial skepticism by others in the industry, industry recognition and praise of the patented products, and efforts by others to copy the patented technology. I am not presently aware of any secondary considerations that would apply here.

43. I have been informed that these secondary considerations of non-obviousness must be balanced against the strength of the prima facie case of obviousness. Where the invention represents no more than the predictable use of prior art elements according to their established functions, secondary considerations are inadequate to establish non-obviousness.

44. I have also been informed that the patentee must establish a nexus between any secondary consideration factors and the claimed invention. Where a secondary consideration factor results from something other than what is both claimed and novel in the patent, there is no nexus to the merits of the claimed invention.

45. I have been informed that, for example, the nexus requirement as it applies to the evidence of commercial success requires that the patentee must show that the sales of a product that allegedly practices the claimed invention must be a direct result of the unique characteristics of the claimed invention, as opposed to

other economic and commercial factors that are unrelated to the patented invention. I understand that this means if the commercial success is due to an element that existed in the prior art or that is not claimed by the patent, then no nexus exists.

46. I have been further informed that, for example, the nexus requirement as it applies to the long-felt need element must likewise be supported by evidence. Where the differences between the prior art and the claimed invention are minimal, it cannot be said that any long-felt need was met by the patented invention or that the patent solved any technological problems that were unaddressed by others.

47. I have been informed that evidence of industry praise must also show a nexus with the claimed invention, and that self-serving statements by the patentee regarding the merits of the claimed invention are generally not given weight.

VI. TECHNOLOGY OVERVIEW

48. Network stacks were well known at the time of the invention of the ’918 patent. The network stack refers to a model of layers for processing network traffic. There are several models of such layers, including the OSI model which originated in the early days of data networking and was published as a standard in 1984.

49. The TCP/IP model originated with the introduction of TCP in 1974. EX1061. When data is received at the lowest layer (or bottom) of a network stack, such as at the link layer (e.g., the Ethernet or MAC layer) the data frames are

processed to determine the next layer up which to pass the frame after removing the current layer's header information. *See, e.g.*, EX1064, §2.3.3. In the Internet, the next layer up is the network layer (also known as the Internet layer), and the most common protocol at this layer is the Internet Protocol, or "IP." *Id.*, §1.1.3. Above the network layer is the transport layer, and the most common protocols at the transport layer in the Internet are TCP and UDP. *Id.* Above the transport layer is the application layer (*Id.*), which include the HTTP protocol for accessing the web and the SIP protocol for initiating a media session. Because the network data is received and processed through these layers, the specific types of protocol processing are known as the network stack, since the layers are stacked on top of each other.

50. It was known as of the Critical Date (*see infra* §IX) of the '918 patent that media objects in the network were referenced via identifiers that included URLs. URLs were introduced when the World Wide Web was invented in the late 1980's and formalized with the publishing of IETF RFC 1738 in December 1994. EX1065. URLs could specify a particular protocol to use, such as HTTP (for accessing a web server), FTP (for accessing a file transfer protocol server), NEWS (for accessing a USENET news server), among other protocols. When the SIP protocol was first published as RFC 2543 in March 1999, the URL protocol specification for SIP was also introduced. EX1066, §2.

51. An HTTP URL is commonly used to access web pages and media files, such as for an image or a video. While images were accessible to access directly on the web by 1992, they were integrated into web pages as inline images in 1993, and images were adopted into the second version of the HyperText Markup Language ("HTML") standard published as RFC 1866 in 1995. EX1067. SIP URLs can be used to initiate communication sessions, such as for a voice call or a video session.

52. Proxy servers were well known at the time of the invention of the '918 patent. Web proxy servers were introduced around 1994, and could be used to cache commonly accessed web objects for an organization, so that Internet bandwidth usage could be reduced by serving requests for data from a local cache server instead of requesting popular content from the original web server. Content Delivery Networks also utilize proxies to cache popular content on behalf of a content provider. A CDN will locate their numerous proxies close to large user bases, and cache content on behalf of its customers so as to be more responsive to user requests.

VII. THE '918 PATENT

A. Specification

53. The '918 patent is directed to "a wireless end-user device" that includes a "proxy network service manager" (also referred to as a "proxy") that facilitates media requests from applications. EX1001, Abstract, 71:21-42, 110:12-111:17, 119:49-60, FIGS. 30, 35. In an embodiment in FIG. 30, the application "utilizes an

API to trigger the proxy 3012 which in turn passes through a socket connection at the socket 3016 as traffic." *Id.*, 110:46-53. In FIG. 32, an embodiment depicts application 3202 communicating through a "control application 3204" that is positioned between the application and proxy service manager 3206. *Id.*, 116:39-118:50, FIG. 32. FIG. 35 depicts another embodiment that includes "a proxy service manager 3502," a "proxy/library API 3504," a "stack API 3518," and a "usage/classification reconciliation engine 3526." *Id.*, 119:49-60.

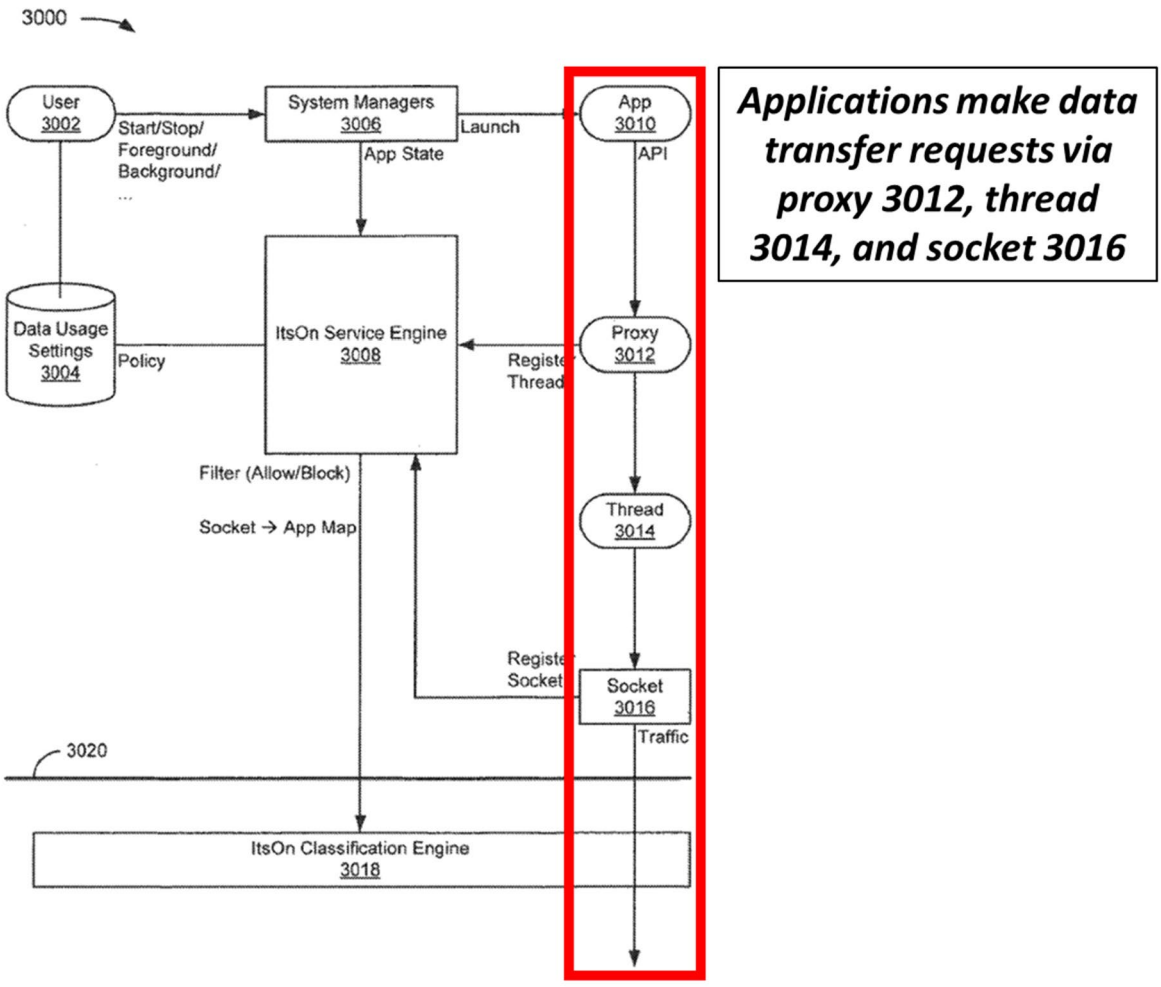


FIG. 30

EX1001, FIG. 30.[1]

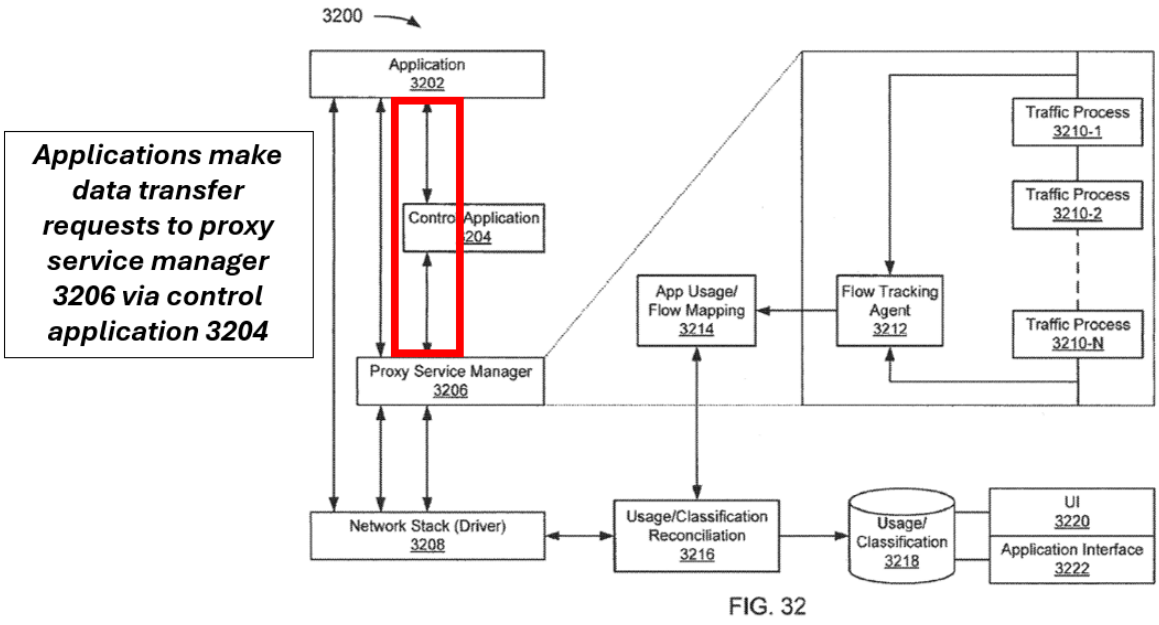


FIG. 32

EX1001, FIG. 32.

[1] Annotations to figures are shown in color.

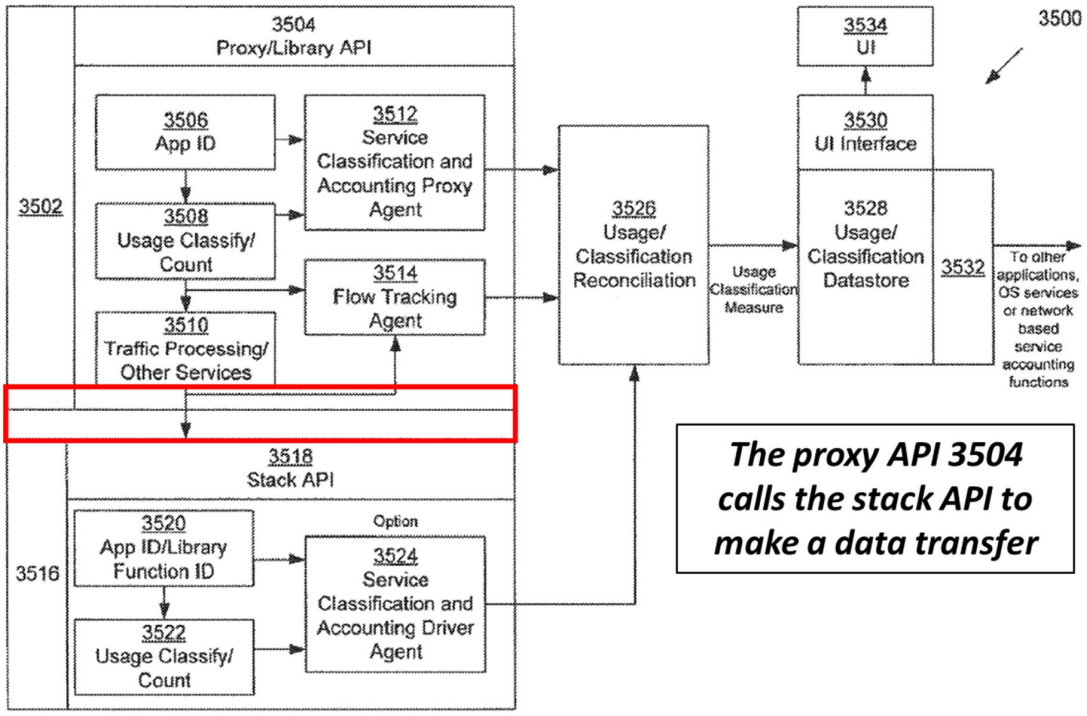


FIG. 35

EX1001, FIG. 35.

B. Prosecution History

54. The Examiner issued only one office action during the '918 patent's prosecution, including prior art rejections of claims 2-4, 11, and 13-16 (issued claims 1-2, 9, and 11-14) under §102 over Deu-Ngoc (US 8,402,165), and a rejection of claim 5 (issued claim 3) under §103 over Deu-Ngoc in view of Constantin of (US 2014/0241342). EX1002, 381-390. The Examiner also indicated that claims 6-10 and 12 (issued claims 4-8 and 10) were allowable. *Id*. The applicant amended the claims to overcome the above rejections and added new claims 17-21 (issued claims 15-19). EX1002, 99-112. All claims were then allowed. EX1002, 21-25.

VIII. CLAIM CONSTRUCTION

55. I understand that, in this proceeding, the claims are construed according to their ordinary and customary meaning, in light of the specification and prosecution history, as understood by a person of ordinary skill in the art at the time of the invention. I understand that this is the same claim construction approach used in district court litigation.

56. I have been instructed that the "intrinsic record" includes the '918 patent itself, including the claims, description, and figures (EX1001), and the '918 patent's prosecution history—i.e., the record of proceedings at the U.S. Patent and Trademark Office ("USPTO") concerning the patent (EX1002). I understand that, like the claims and written description, the prosecution history provides evidence to a person of ordinary skill in the art of how the inventor intended his patent to be understood, and how the USPTO understood the patent. I understand that the inventor is permitted to apply a special definition to the terms or to limit the scope of claim terms in his patent claims, which may differ from the term's plain and ordinary meaning. That special definition or limitation on scope may be provided in the patent's written description, the patent's prosecution history, or both.

57. I understand that claim interpretation may also be informed by "extrinsic evidence" (that is, evidence outside of the patent record itself). I have been informed that extrinsic evidence may include dictionaries, technical treatises, and

other materials evidencing the meaning of a claim term and the understanding held by a person of ordinary skill in the art in the relevant time period.

58. In my opinion, none of the claim terms in the '918 patent require an express construction. Instead, the claims should all be given their plain and ordinary meaning as they would have been understood by a person of ordinary skill in the art at the Critical Date (*see infra* §IX) of the '918 patent. I have been informed that Patent Owner has previously asserted that the prior art does not teach claim 1's "first call accessible to each of a plurality of device applications," and "API callable by each of the plurality of device applications" because intermediaries are interposed between the APIs and applications. As I discuss in greater detail below, this interpretation is inconsistent with the plain and ordinary meaning of the term, which permits indirect access between the applications and APIs. *See* §§X.A.4.d (1[c]), X.A.4.e (1[d]).

IX. LEVEL OF ORDINARY SKILL

59. I am informed that patentability must be analyzed from the perspective of "one of ordinary skill in the art" in the same field as the patent-in-suit at the time of the invention. I am also informed that several factors are considered in assessing the level of ordinary skill in the art, including (1) the types of problems encountered in the art; (2) the prior art solutions to those problems; (3) the rapidity with which innovations are made; (4) the sophistication of the technology; and (5) the education

level of active workers in the field. I have been asked to assume that the priority date (or Critical Date) of the '918 patent is January 24, 2011.

60. A person of ordinary skill in the art ("POSITA") at the '918 patent's effective filing date would have at least a bachelor's degree in computer science, computer engineering, electrical engineering, or a similar field, and approximately two years of industry or academic experience in wireless communication network applications and software. More related education could compensate for fewer years of work experience (and vice versa).

X. UNPATENTABILITY OF THE CHALLENGED CLAIMS

A. Ground 1: *Bennett* and *Vadde* render Claims 1-3, 8-9, 13-14, and 19 Obvious

1. Overview of *Bennett*

61. *Bennett* discloses a "media client" including a "user agent to communicate with a multimedia application in the networked communication device," a "signaling agent … to establish and maintain communication sessions," and a "media agent" which "performs media operations." EX1041, Abstract, ¶¶24-26, 29-31, FIG. 3. Media operations include "Push-to-Talk over Cellular (PoC), presence and Instant Messaging (IM), video and audio streaming, voice over IP videoconferencing, interactive gaming, white-boarding and content sharing." EX1041, ¶24. *Bennett* discloses that its media client is implemented in a "mobile device" that includes the "[user agent] 202, [signaling agent] 204 and [media agent]

206." EX1041, ¶¶28, 78. *Bennett*'s media agent "stream[s]" media to a "media player" on the device, which outputs the media to the user using "a local media rendering device (e.g., speaker and/or display of a mobile terminal 100)." EX1041, ¶¶25, 76, FIG. 3. Additionally, *Bennett* discloses that media can also be routed directly to the application. ¶76, FIG. 10.

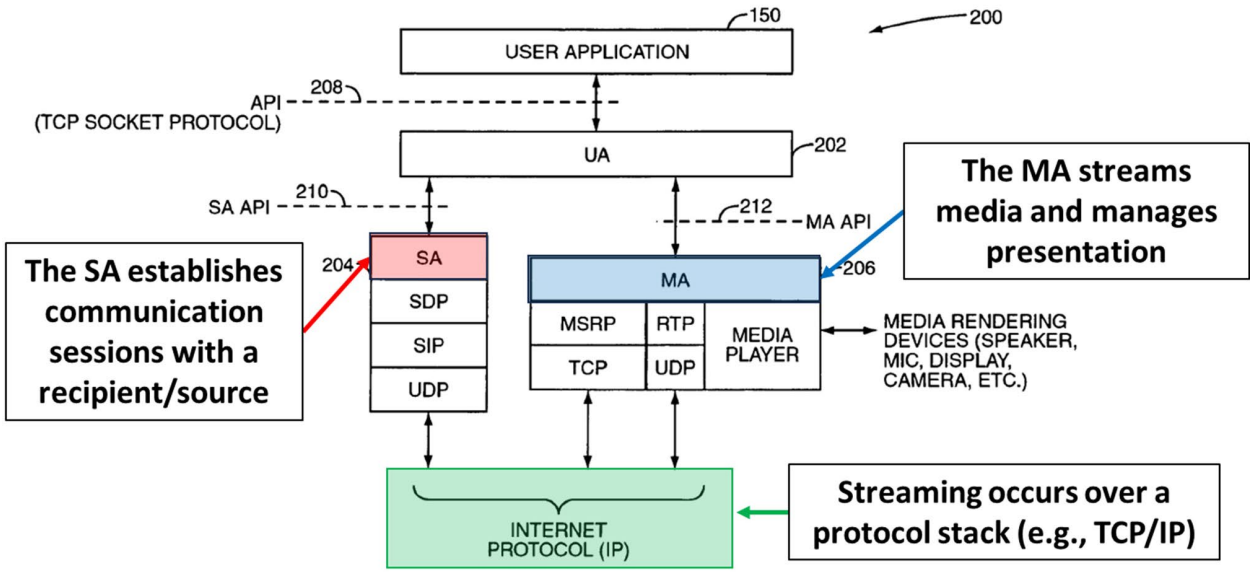


FIG. 3

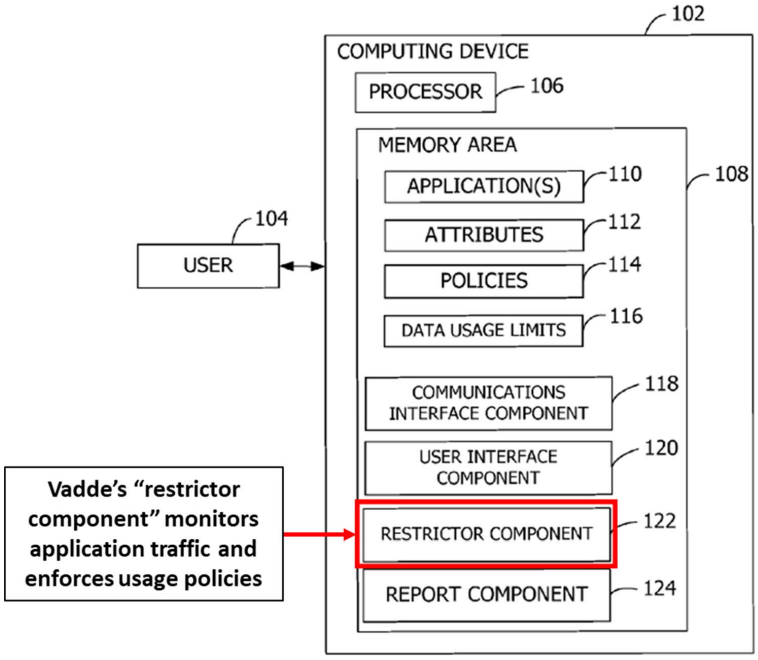
EX1041, FIG. 3

2. Overview of *Vadde*

62. *Vadde* discloses techniques for "managing data traffic" for a plurality of applications on a "computing device 102." EX1042, ¶¶9-23. *Vadde* enforces restrictions on application data usage based on "attributes" and "usage limits." EX1042, ¶¶15-16, 25-26. *Vadde*'s techniques are applied for "each" application and

are enforced with a "restrictor component 122" which "appl[ies] [a] data usage policy" and "monitors the data transmitted and/or received by the applications 110 and determines whether the data usage limits 116 corresponding to each of the applications 110 have been exceeded or are about to be exceeded." EX1042, ¶¶16, 22. An example computing device is depicted below. EX1042, FIG. 1.

FIG. 1



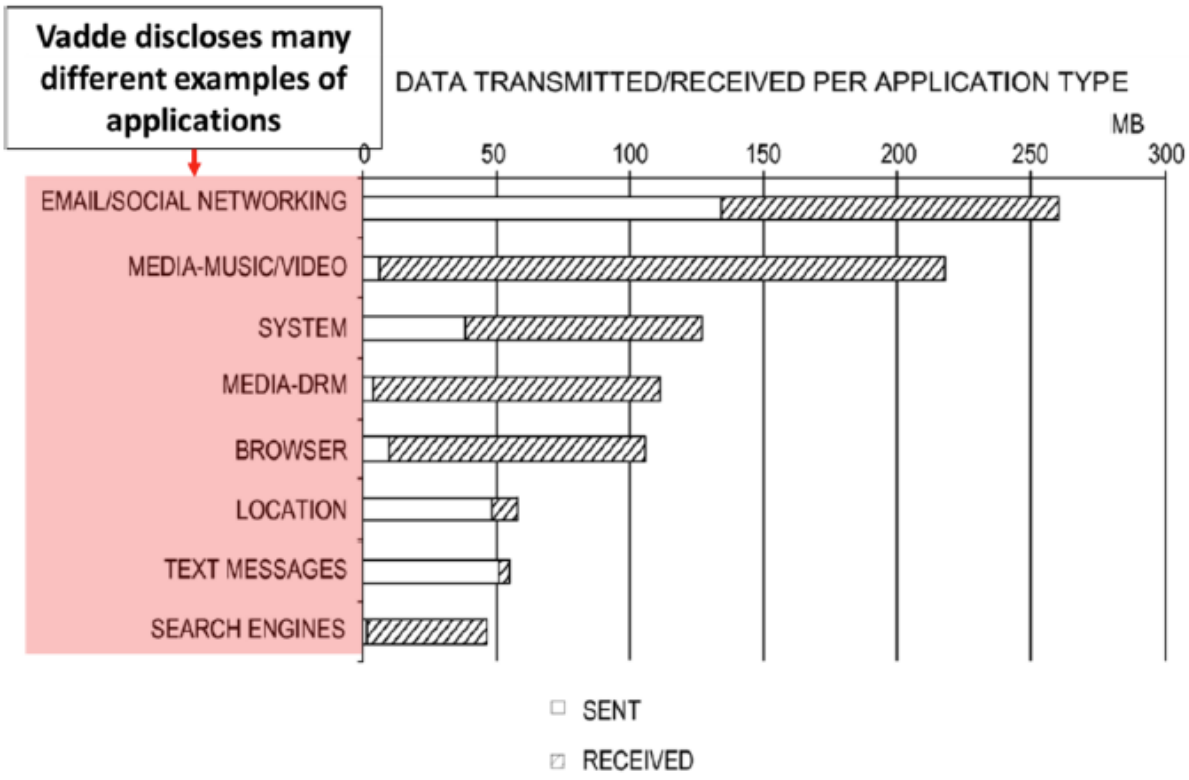
EX1042, FIG. 1.

3. Motivation to Combine *Bennett* and *Vadde* with Reasonable Expectation of Success

63. It would have been obvious for a POSITA to incorporate *Vadde*'s techniques of managing data traffic—to include *Vadde*'s policies and restrictor component—into the media clients of *Bennett* to monitor data usage for applications

making "media object network data transfers" and using "respective data packet flows." EX1041, ¶78; EX1042, ¶¶10, 22. In my opinion, a POSITA would have been motivated to make this combination for multiple reasons.

64. First, as *Vadde* notes, "[t]he cost of mobile operator data plans is often based on usage" and "with existing systems, users are unable to determine the relative costs incurred by different applications executing on a mobile telephone." EX1042, ¶1. Further, Vadde discloses that many various different applications of different types (to include "media-music/video") can be operating on a mobile device. EX1042, ¶31, FIG. 4. Incorporating *Vadde*'s data usage monitoring into the *Bennett* device would have enabled the user of the *Bennett* device to monitor their data usage and avoid unforeseen expenses. *Id*. The ability to identify and restrict data-intensive applications would have also allowed a user and network provider to "reduce battery usage" caused by these applications, extending operating time between charging. EX1042, ¶¶9, 23-24.



EX1042, FIG. 4.

65. Indeed, the benefits of data usage monitoring were well-known prior to the '918 patent and would have been known to a POSITA. For example, *Rybak* corroborates the need to monitor data usage and discloses a "method for monitoring resource usage of a mobile communications device" with respect to a "mobile communication plan profile associated with a subscriber." EX1044, ¶¶3-4, 30-42, 47, FIGS. 2, 7. Like *Vadde*, *Rybak* notes that resource usage correlates to a "billing period" or "plan" that would be chargeable to a user. EX1044, ¶¶31-34, 45, 60, FIGS. 7-8. *Benco* discloses another example of a "method for providing mobile telephone subscribers with data on accumulated usage" where "[s]ubscribers are

warned if their accumulated usage threatens to exceed or exceeds the allowable basic usage of their billing plan." EX1045, Abstract, ¶¶1-20. *Jobs* provides another example, where a user interface of a mobile device "displays an updated account usage metric for an account associated with usage of the device (e.g., a cellular phone account)." EX1062, ¶213. *Fadell* is yet another example of "metering" network resource usage on a mobile device to prevent a user from exceeding a "resource allocation." EX1063, ¶¶2-7, 17, 50-53, 56. With the known benefits of data usage monitoring, a POSITA would have been motivated to consider and include *Vadde*'s data usage monitoring in *Bennett*'s device.

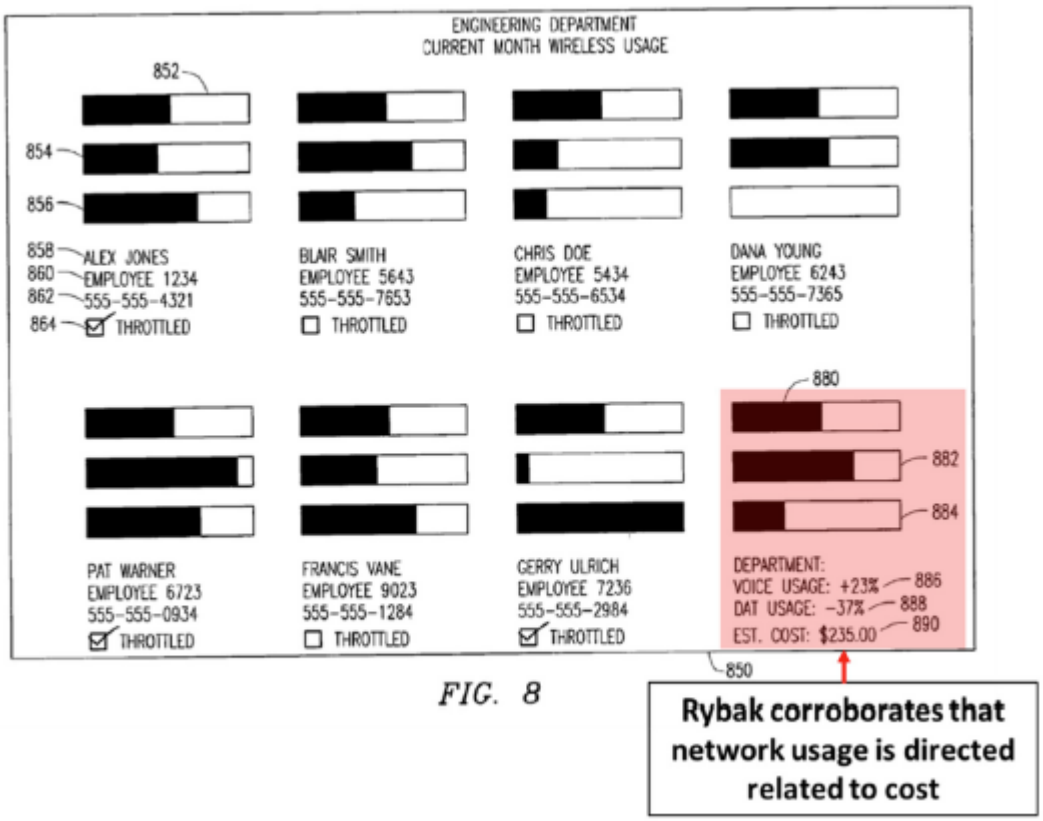


FIG. 8

EX1042, FIG 8.

66. Second, because *Vadde*'s policies are applied at an application level, *Vadde*'s techniques would have enabled the *Bennett* service providers greater control over specific application activity that was data-intensive for the network (e.g., streaming video while roaming). EX1042, ¶¶10, 16, 18, 22, 25, 29-30, 32. These restrictions would have enabled service providers to prevent an aggregation of data-intensive activity from degrading a network's effectiveness. *Id*. For example, *Vadde* discloses that mobile devices can be an "employer-sponsored mobile telephone," and an employer would typically have great interest in how their company devices were being used at the user level. EX1042, ¶¶33-34. Moreover, "roaming" may also be extremely financially costly for the user, particularly for high-bandwidth activities like streaming, and was typically avoided if possible. EX1042, ¶34; EX1046, 7:40-50, 14:61-67, Table-2 (describing "real-time audio and video" applications as "[h]igh bandwidth" applications).

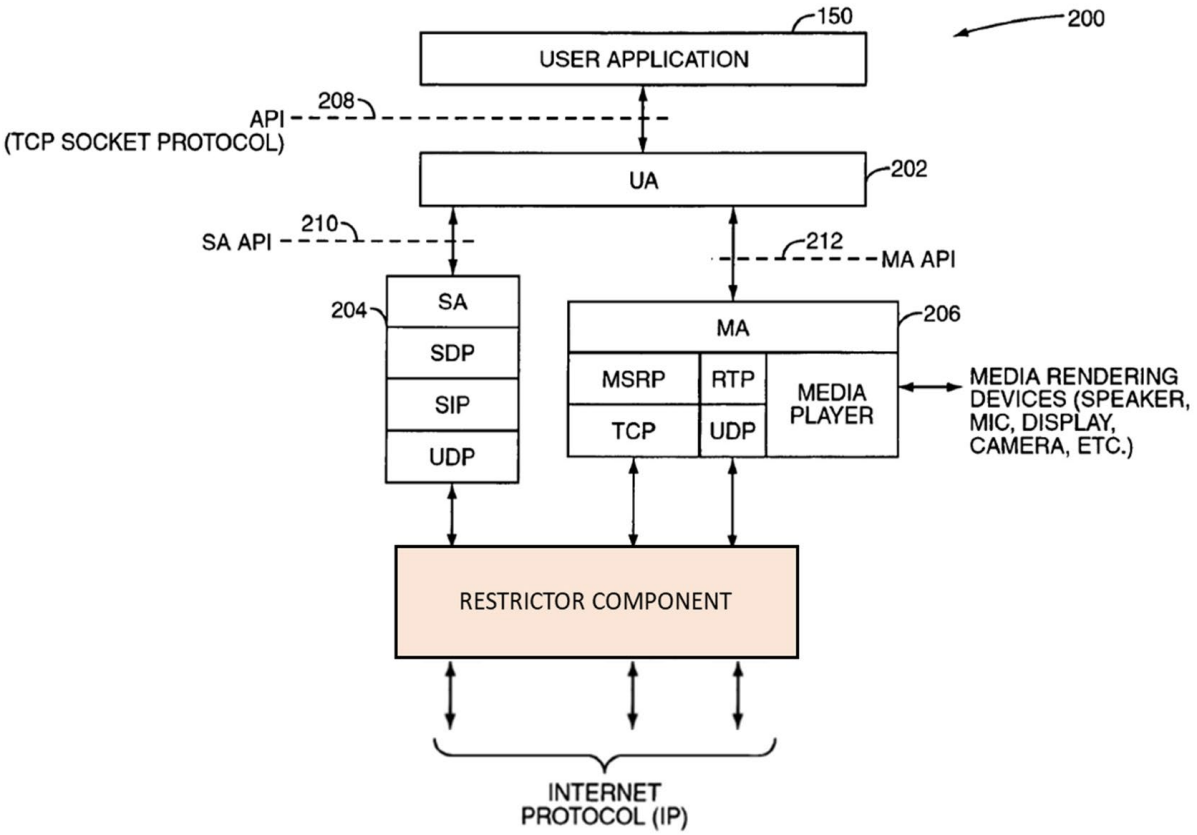
67. Finally, adding per-application data usage monitoring to the *Bennett* device would have enabled device manufacturers and service providers to incorporate additional applications and functionality into wireless devices while allowing users to retain control of aggregate device data usage. EX1042, ¶¶10, 16, 18, 22, 25, 29-30, 32. Indeed, *Bennett* acknowledges that "[t]he convergence of mobile and IP networks will allow service providers to offer ***new IP services*** to mobile subscribers that were previously available only to users in fixed networks,

such as the Internet." EX1041, ¶2. *Vadde*'s techniques would have allowed service providers to incorporate these "new IP services" described by *Bennett* while retaining control of device wireless data usage in higher-cost wireless networks. EX1041, ¶2; EX1042, ¶¶10, 16, 18, 22, 25, 29-30, 32.

68. Incorporating *Vadde*'s techniques into the *Bennett* device would have been nothing more than the application of known techniques (e.g., managing data traffic according to *Vadde*) to a known device (e.g., *Bennett*'s devices) to yield predictable results (e.g., the management of the *Bennett-Vadde* device's data traffic). In my opinion, a POSITA would have expected success in implementing this combination because the monitoring of outgoing API data traffic on a mobile device (such as the UA, SA, and MA APIs of *Bennett*) was known as of the Critical Date and would have involved only routine programming skill.

69. A POSITA would have been motivated to monitor application data usage in *Bennett* using *Vadde*'s restrictor component to determine data usage through *Bennett*'s SA API and MA API so that aggregate traffic data, regardless of which API is used, would be captured on a per-application basis and compared to data usage limits. As explained above, tracking an aggregate application data usage on a mobile device was known, EX1044, ¶¶3-4, 30-42, 47, FIGS. 2, 7; EX1045, Abstract, ¶¶1-20, and it would have been obvious to track application data usage via *Bennett*'s SA API and MA API in the combined system because *Bennett* discloses

these are the two pathways applications request data, EX1041, ¶25. This would have fulfilled *Vadde*'s goal of "manag[ing] data traffic including data transmitted and/or received by ***each of a plurality of applications*** executing on a computing device," and allow applications using significant data to be identified to avoid exceeding data limits. EX1042, Abstract, ¶¶1-3, 22-23. *Rakoshitz* supports this by disclosing a method for "monitoring or profiling quality of service within a network of computers" that includes a "policy engine module" that "interfaces with [an] API" to enforce a "traffic policy [that] defines specific limitations or parameters for the traffic." EX1046, Abstract, 9:18-24, 12:12-58, Claim 1. *Michels* provides another example of a system for "monitoring and control of access to [an] API" with a processor that "monitors the distribution of the API elements" and "the number of API requests made by [a] developer client over a period of time, the identity of the developer client, usage trends by the developer client, and usage trends based on IP address." EX1049, ¶¶2-15, 43-63. Furthermore, as explained above, tracking an aggregate application data usage on a mobile device was also known. EX1044, ¶¶3-4, 30-42, 47, FIGS. 2, 7; EX1045, Abstract, ¶¶1-20.



EX1042, FIG. 3 (as modified by *Vadde*).

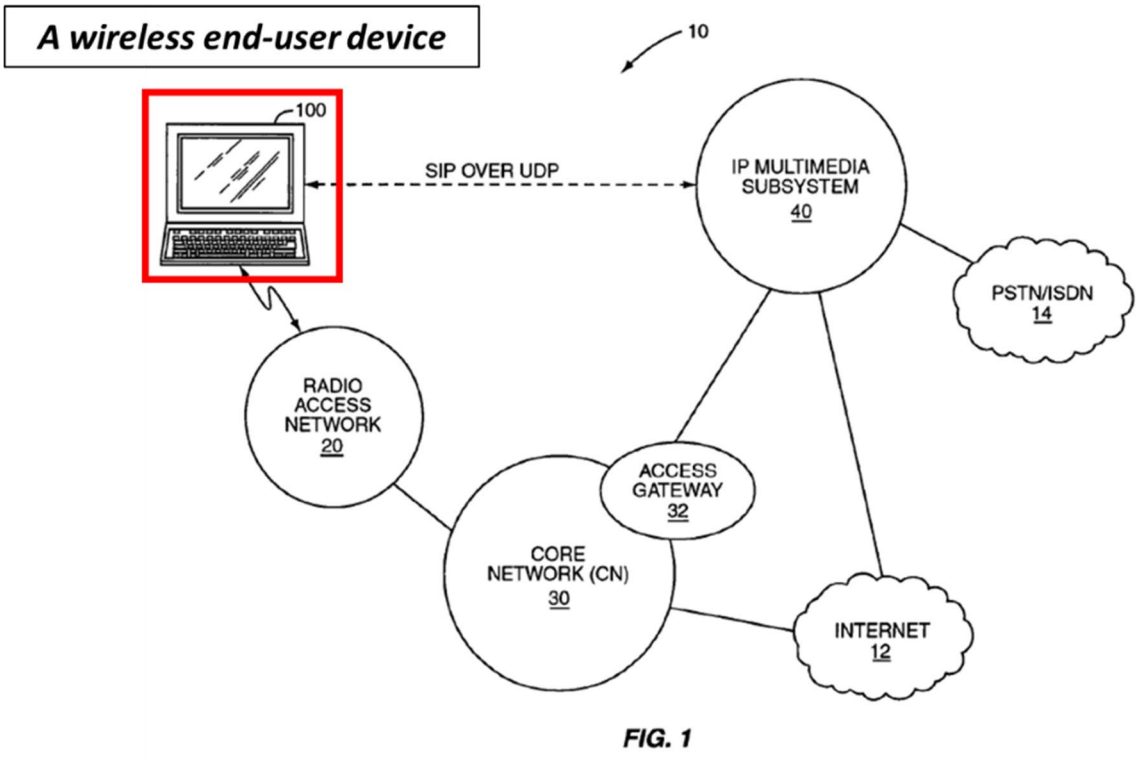
4. Claim 1

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

70. The '918 patent does not define a "wireless end-user mobile device," but instead describes various devices that can implement its techniques, to include "***mobile devices***, such as phones, PDAs, computing devices, laptops, net books, tablets, ***cameras, music/media players***, GPS devices, networked appliances, and any

other networked device." EX1001, 40:28-46.[2] The prior art here discloses similar devices.

71. To the extent the preamble is limiting, *Bennett* discloses that its techniques can be implemented in a "mobile device," "video camera," and "remote video player" (all "***wireless end-user device[s]***" which communicate over wireless networks). EX1041, ¶78; *see also* ¶¶2, 5, 25, FIGS. 1, 4, 11.



EX1041, FIG. 1.

[2] All emphasis is added unless otherwise noted.

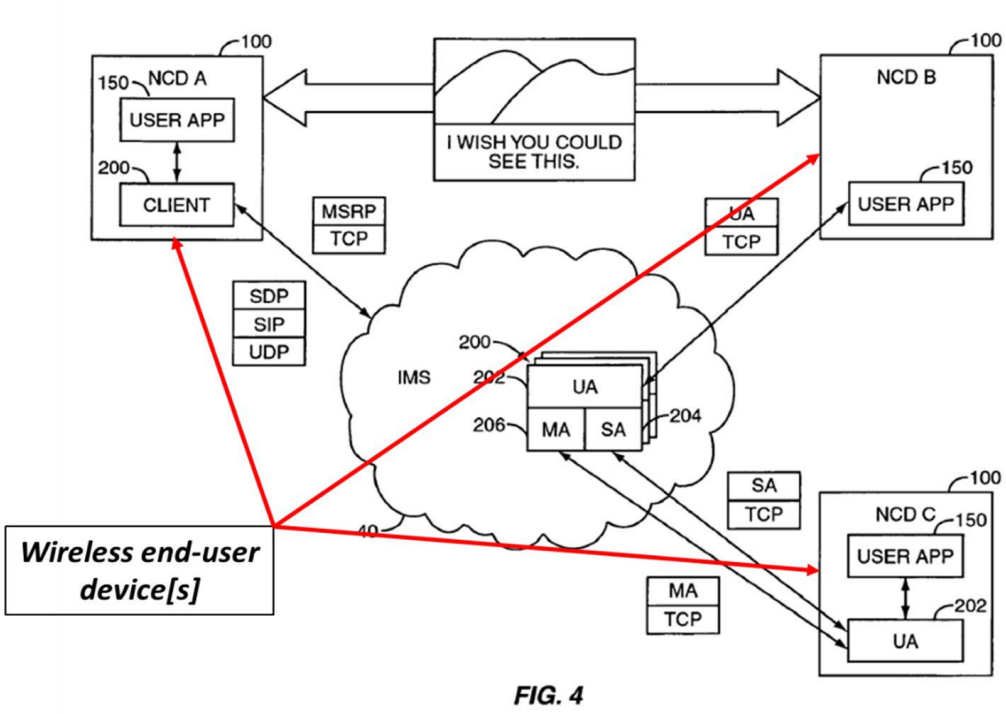


FIG. 4

EX1041, FIG. 4.

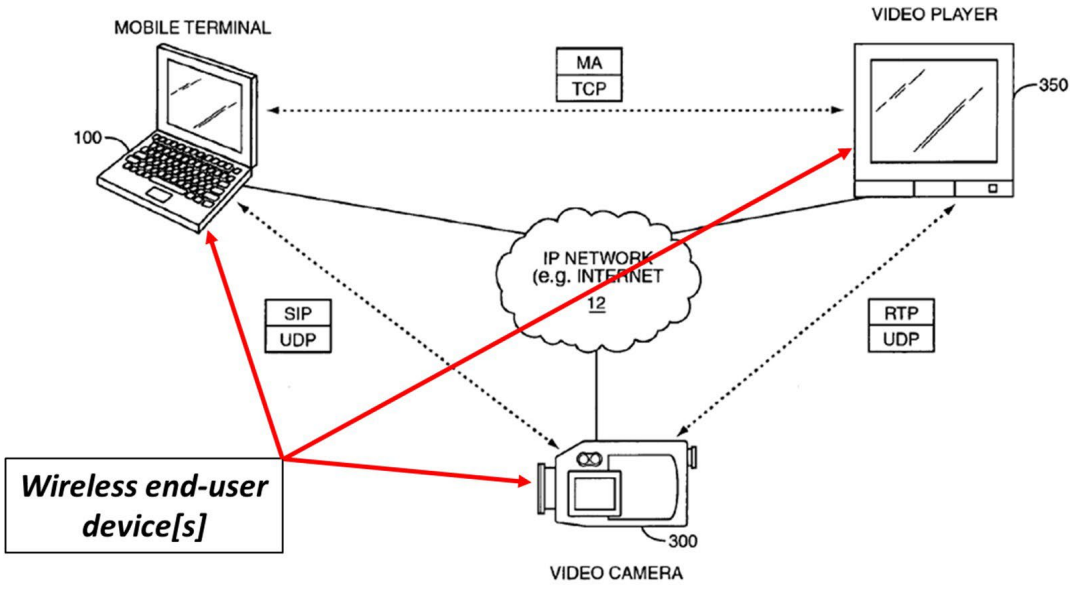


FIG. 11

EX1041, FIG. 11.

72. *Bennett* also discloses a "media client 200" including a "user agent to communicate with a multimedia application in the networked communication device," a "signaling agent … to establish and maintain communication sessions," and a "media agent" which "performs media operations." EX1041, Abstract, ¶¶24-26, 29-31, FIG. 3. *Bennett*'s media client 200 is entirely contained within a "***wireless end-user device***." EX1041, ¶¶28, 78, FIGS. 4, 11.

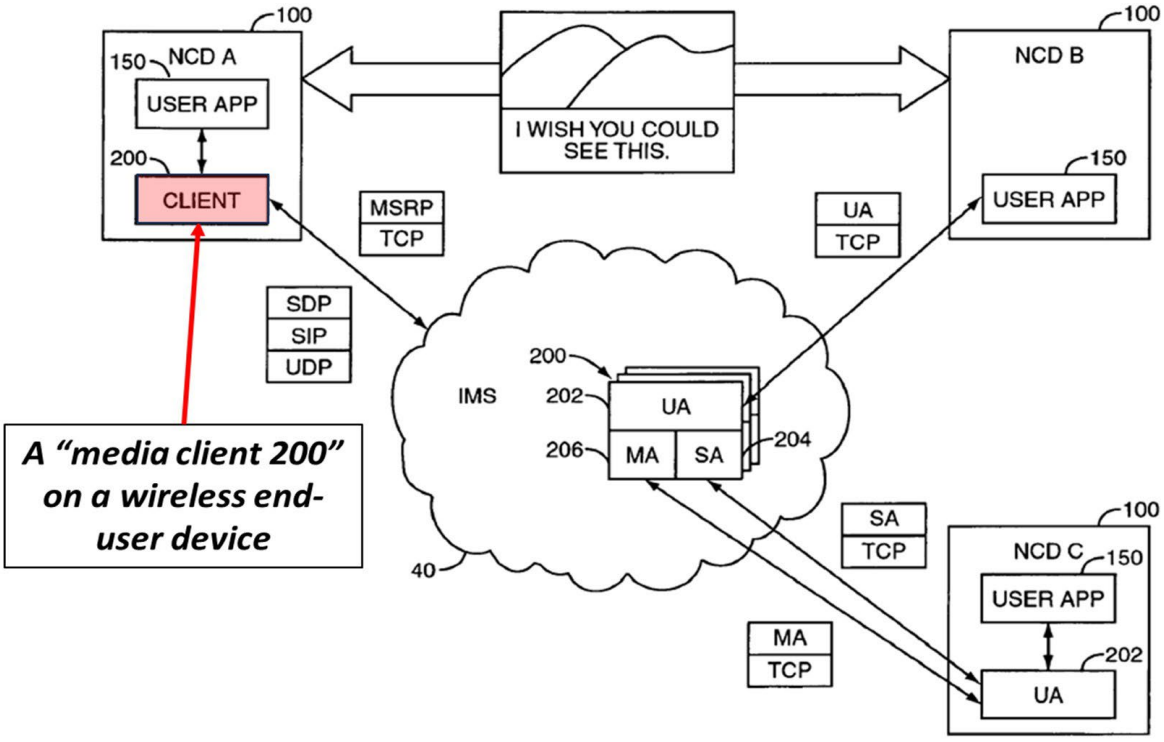


FIG. 4

EX1041, FIG. 4.

b. 1[a]: *"a wireless modem configurable to connect to a wireless network"*

73. *Bennett* discloses that its mobile devices operate on various "mobile communication network[s]." EX1041, ¶17, FIG. 1. *Bennett*'s networks include "General Packet Radio Services (GPRS) network[s]" and Universal Mobile Telecommunications Service ("UMTS") networks ("***wireless network[s]***"). *Id*.

74. A POSITA would have understood and found obvious that to communicate and retrieve media via these networks, *Bennett*'s mobile devices would have had the ability to modulate and demodulate data using a "***wireless modem***." Indeed, it was well known before the Critical Date that "***wireless modem[s]***" were used to transmit data from mobile devices operating on wireless networks, including GPRS and UMTS networks, and that these modems were "***configurable to connect to a wireless network***." EX1047, ¶¶25, 68. For example, *Cassett* discloses that devices communicating in "GPRS" and "UMTS" networks included "wireless modems." *Id*.

75. Additionally, a POSITA would have understood and found obvious that mobile devices at the time of the Critical Date typically included "***wireless modem[s]***" that were "***configurable to connect to a wireless network***" as evidenced by multiple prior art references. EX1048, ¶¶34-35, FIG. 2; EX1013, ¶¶125, 130. Indeed, *Cole* discloses a "mobile device" that includes a plurality of "***wireless modem[s]***," to include a "WWAN modem 230," a "WLAN modem 235," and a

"voice band modem 250." EX1048, ¶¶34-35, FIG. 2. *Rao* discloses that "computing device[s] 102" included "network interface[s] 118," for example, a "modem." EX1050, ¶¶125, 130.

c. 1[b]: *"a network stack configurable to receive and transmit data via the wireless modem and the wireless network"*

76. *Bennett* discloses that the media agent 206 is in communication with several "protocol stack[s]" ("***a network stack***") which are configured to transmit and obtain—or "stream"—"media" ("***configurable to receive and transmit data via the wireless modem and the wireless network***"). EX1041, ¶¶18, 25, 60, 66, 75-76, FIGS. 3-4, 8-12.[3]

77. As described above, communication over wireless networks on the *Bennett* mobile end-user device occurs via "***the wireless modem and the wireless network***." *See supra* §X.A.4.b. (1[a]).

[3] *Bennett*'s figures all denote "media client 200," thus indicating the same embodiment. EX1041, FIGS. 3-4, 8-12.

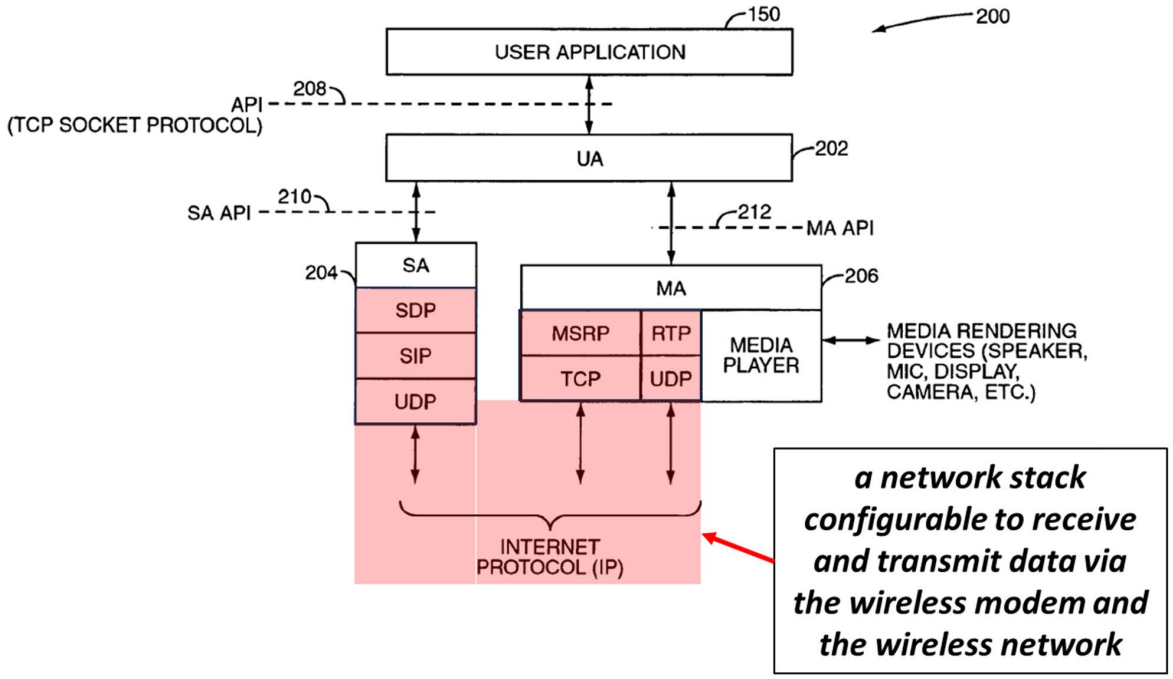


FIG. 3

EX1041, FIG. 3.

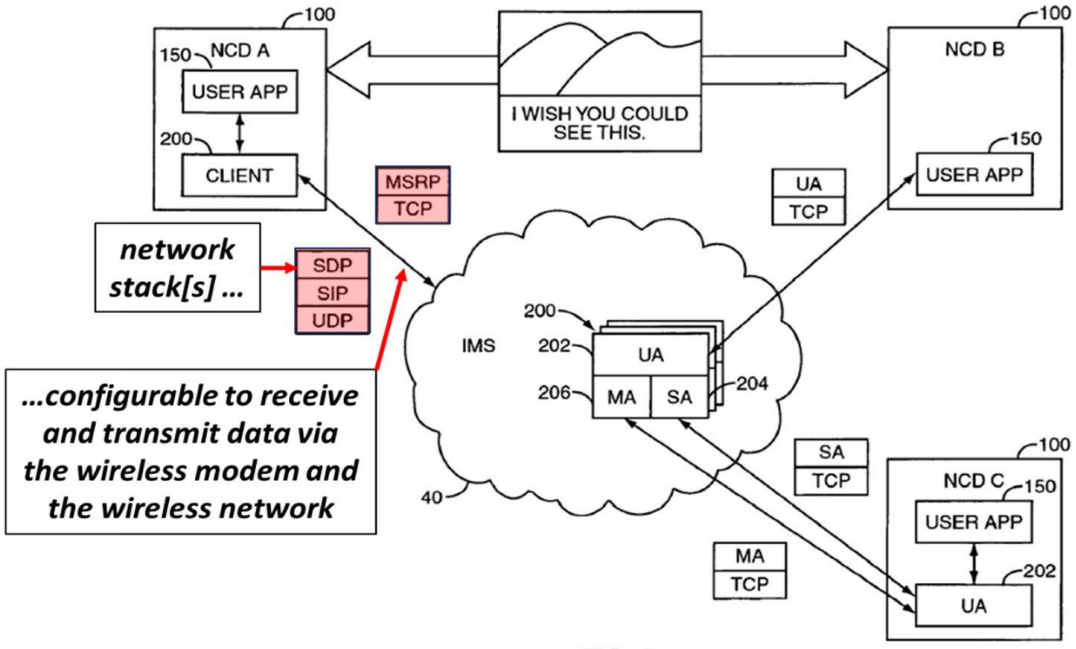


FIG. 4

EX1041, FIG. 4.

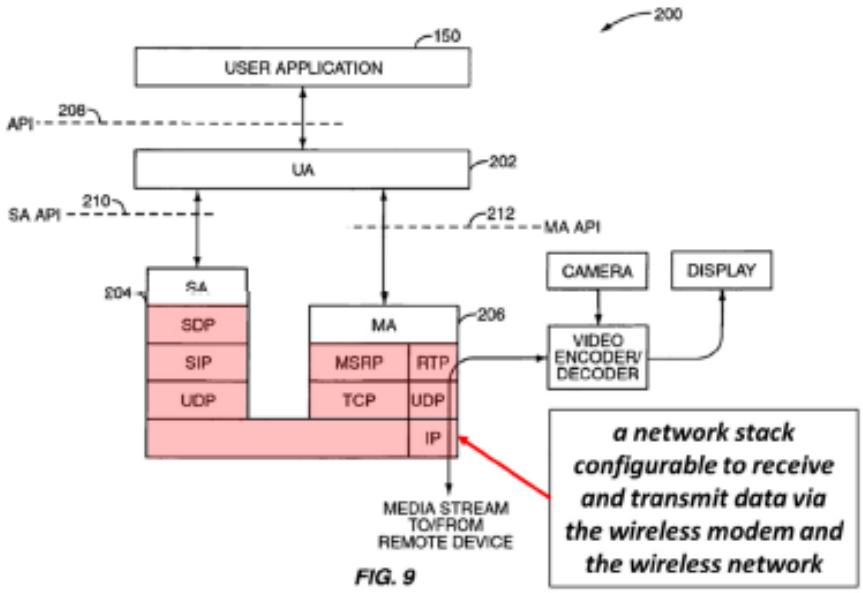


FIG. 9

EX1041, FIG. 9.

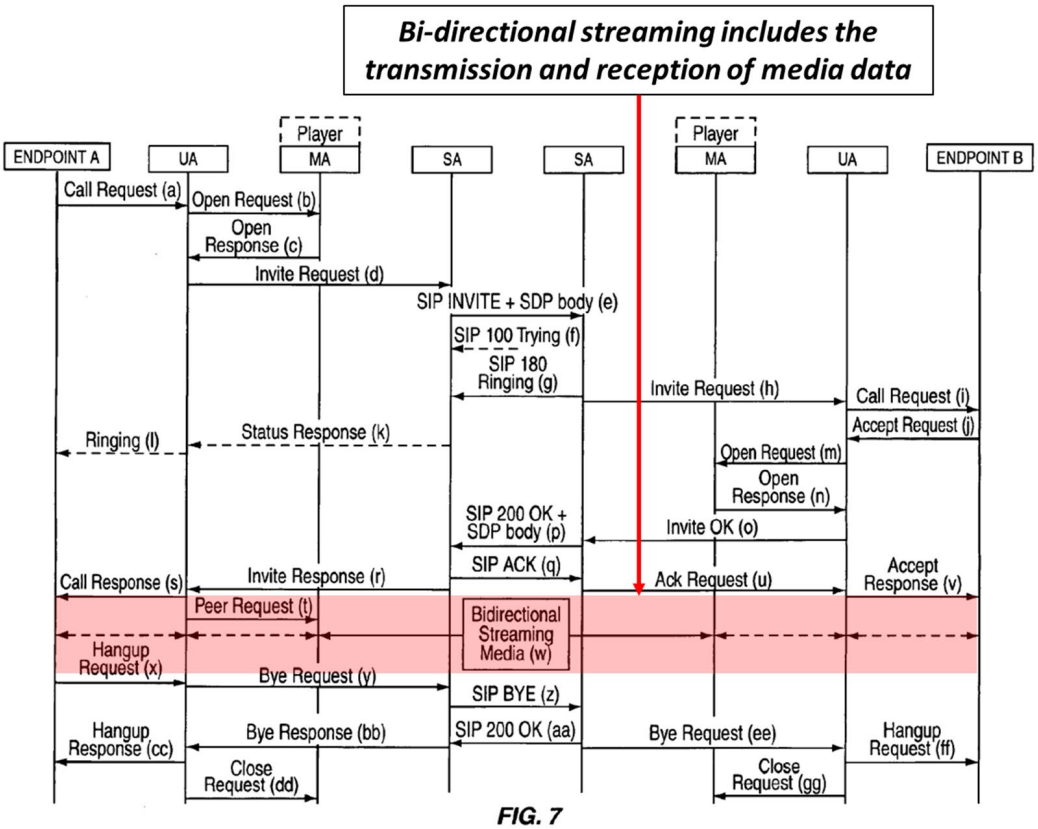


FIG. 7

EX1041, FIG. 7.

78. Similarly, the '918 patent depicts Internet Protocol ("IP"), Transmission Control Protocol ("TCP"), and User Datagram Protocol ("UDP") as part of a "device stack." EX1001, 2:21-26, FIGS. 12-13. A POSITA would have understood that the device stack comprises a "network stack," as this term is known in the industry (*see* §VI discussing how network stack refers to a model of layers for processing network traffic) and as the device stack as pictured and described in the '918 patent shows. *See* EX1001, 2:21-26, FIGS. 12-13. The '918 patent also frequently refers to "IP" techniques with respect to the "network stack." EX1001, 61:51-62, 109:22-26, 112:62-113:9.

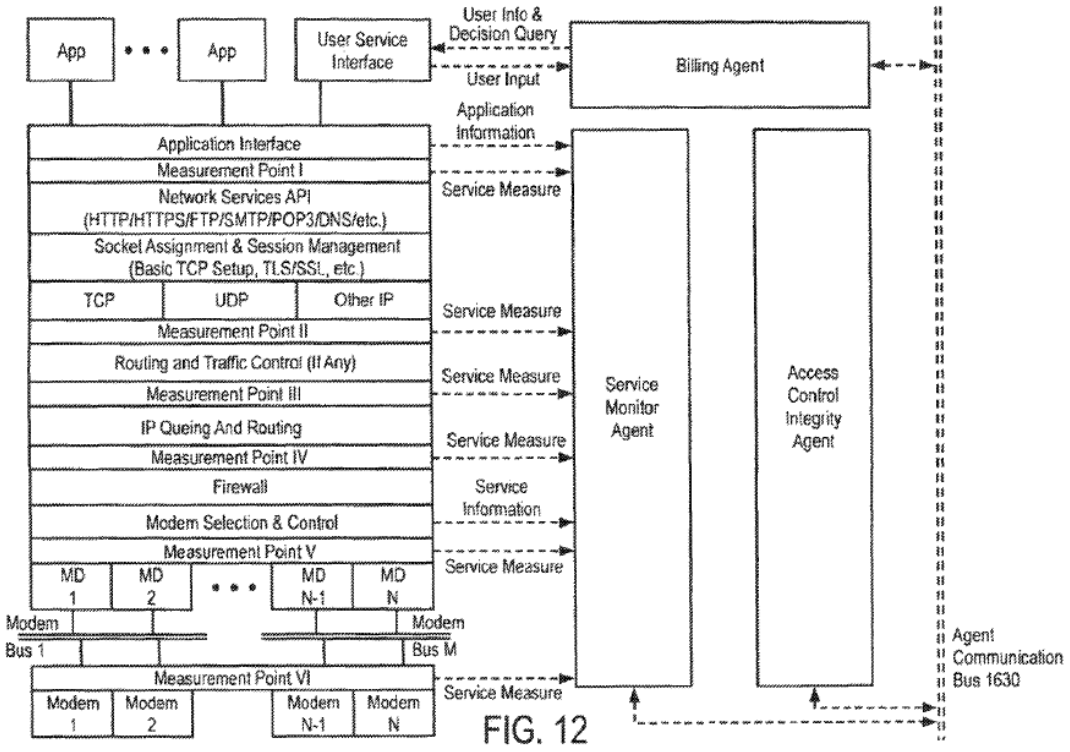


FIG. 12

EX1001, FIG.12.

d. 1[c]: *"a first network stack Application Programming Interface (API), containing at least one first call accessible to each of a plurality of device applications, the first network stack API callable by each of the plurality of device applications to open and use data packet flows via the network stack, the wireless modem, and the at least one wireless network"*

"open and use data packet flows via the network stack, the wireless modem, and the at least one wireless network"

79. *Bennett* discloses the use of "Session Initiation Protocol (SIP)" for "establishing, modifying and terminating communication sessions between one or more participants." EX1041, ¶¶18-22. *Bennett* discloses that SIP enables applications residing on the mobile terminal to "establish a communications session" ("***open … data packet flows via the network stack, the wireless modem, and the at least one wireless network***"). EX1041, ¶¶20-22. SIP sessions in *Bennett* "support multi-media applications," including "Internet multimedia conferences, Internet telephony calls, and multimedia distributions" that are performed using protocols such as "Real-time Transfer Protocol (RTP)" and "Message Session Relay Protocol (MSRP)" ("***us[ing] data packet flows via the network stack, the wireless modem, and the at least one wireless network***"). EX1041, ¶¶20-22 (describing the SIP sessions as being "used for the communication" between parties). In one embodiment, *Bennett* discloses a media client 200 with a signaling agent, SA 204, that "implements SIP and SDP protocols to handle signaling tasks." EX1041, ¶25.

The SA 204 "uses UDP over IP for transport of messages," including "setting up, modifying, and tearing down communication sessions." *Id.*

80. A POSITA would have recognized and found obvious that the protocols disclosed in *Bennett* would have included "***data packet flows***" as these protocols are examples of using "packet switched services" that communicate data in a series of data packets. EX1041, ¶17. Additionally, a POSITA would have recognized and found obvious that the communication of these "***data packet flows***" would have been "***via the network stack, the wireless modem, and the at least one wireless network***" because *Bennett*'s media, included in the data packet flow, is retrieved over various wireless networks, described above, that use protocol stacks and wireless modems. *See supra* §§X.A.4.b. (1[a]), X.A.4.c. (1[b]).

"a first network stack Application Programming Interface (API), containing at least one first call accessible to each of a plurality of device applications, the first network stack API callable by each of the plurality of device applications"

81. As discussed, *Bennett*'s signaling agent (SA) 204 within its "media client 200" that "implements SIP and SDP protocols to handle signaling tasks." EX1041, ¶¶25, 31, 33, 40, 43-49, Table-2, FIGS. 3-10. The SA 204 is called by the "user agent (UA) 202" of the media client 200 using a "SA API 210" ("***a first network stack Application Programming Interface (API)***") in response to a request from a "user application 150" ("***the first network stack API callable by each of the plurality of device applications***"). EX1041, ¶¶31-33, 40, 43-49.

82. The SA API 210 is an API that provides access to the network stack (*see supra* §X.A.4.c (1[b])) that provides access to a set of protocols enabling communication over a network. For example, SA API 210 provides access to the signaling agent 204 that "implements SIP and SDP protocols to handle signaling tasks" and "uses UDP over IP for transport of messages." *Bennett* ¶25, FIG. 3. SA API 210 thus discloses a first network stack API.

83. *Bennett* discloses that "the application interface between UA 202 and SA 204 (the SA API) … use text-based interface protocols similar to the UA API." EX1041, ¶31 (referencing "Table 2 in Appendix B documents exemplary requests and responses for the SA API"). The user application 150 "typically sends requests to the UA 202 to initiate a transaction," where the requests "typically have parameters, which are separated by spaces." EX1041, ¶30. Requests from the user application via UA 202 "requiring action by … the SA 204 … start a transaction." EX1041, ¶31. In one exemplary request ("***at least one first call accessible to each of a plurality of device applications***"), the user application, via UA 202, requests the SA 204 to perform actions in SIP sessions, including "INVITE" requests which identify a source of data or a recipient. EX1041, ¶¶31, 33, 40, 45-46, FIGS. 6-7, Table-2 (disclosing API messages, parameters, and parameter descriptions for communication via SA API 210). This example, as well as those disclosed in Table

2, disclose "***at least one first call accessible to each of a plurality of device applications***."

84. *Bennett* frequently refers to "application**s**" operating on its devices and says that "any" application 150 can use its techniques ("***each of a plurality of device applications***"). EX1041, ¶¶18, 22, 29, 75. *Vadde* also explicitly describes performing its techniques for "a plurality of applications executing on a computing device" ("***a plurality of device applications***"). EX1042, Abstract, ¶¶2, 43.

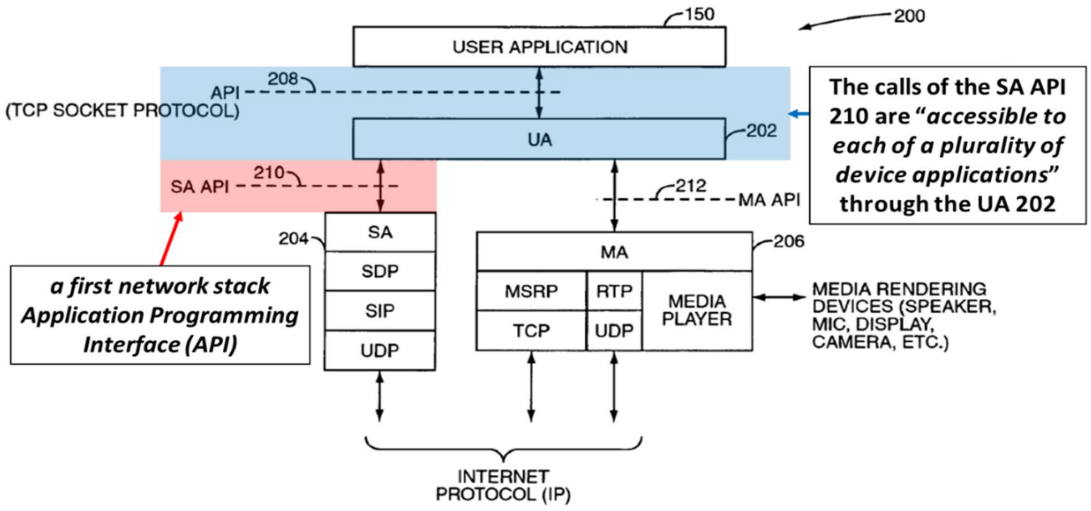


FIG. 3

EX1041, FIG. 3.

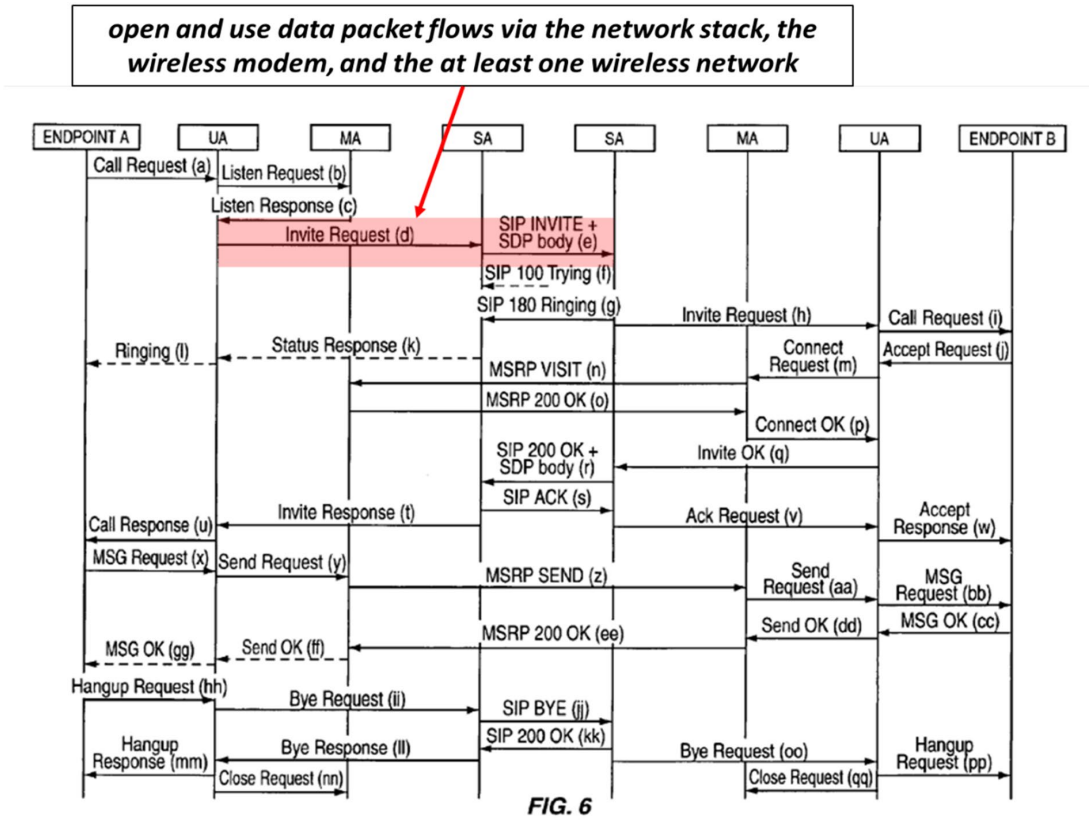


FIG. 6

EX1041, FIG. 6.

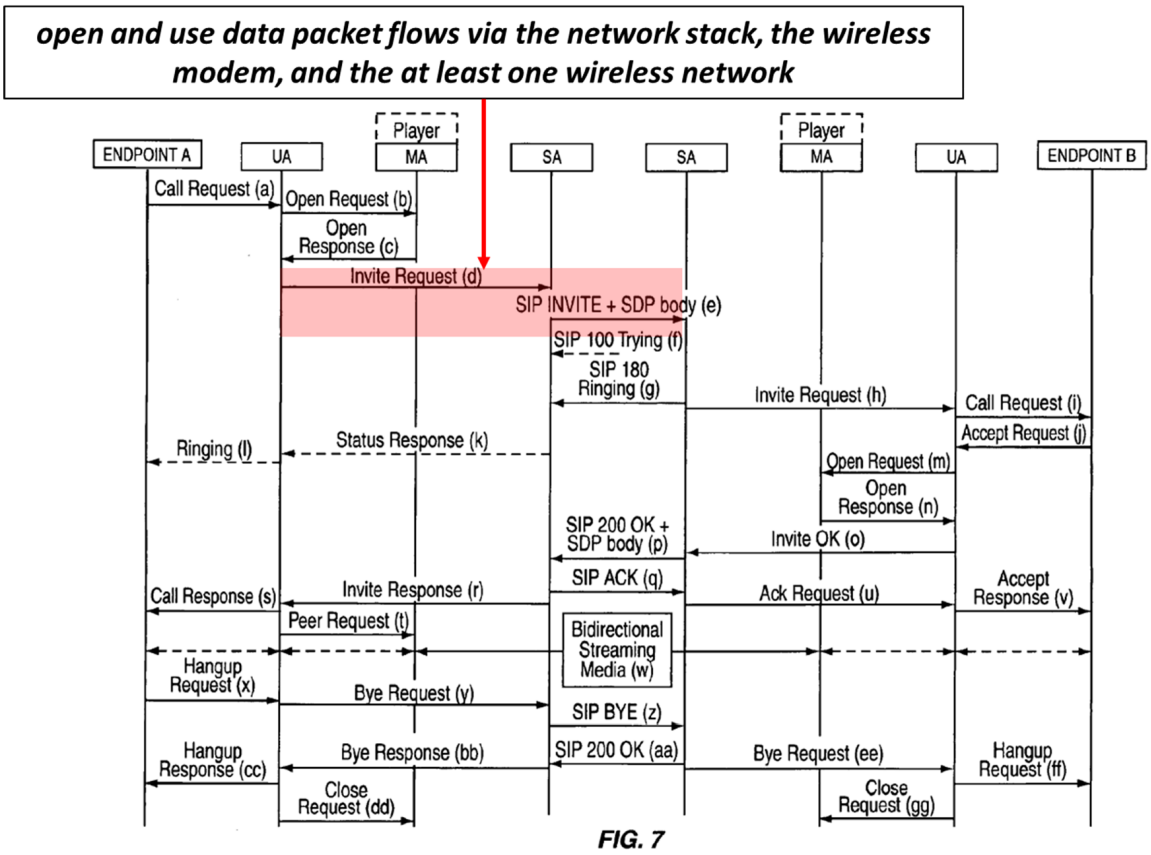


FIG. 7

EX1041, FIG. 7.

85. To the extent there is a question that the calls of the "***first network stack Application Programming Interface (API)***" must be directly "***accessible to each of a plurality of device applications***," or that the application must directly call the "***first***" API, the '918 patent does not support such a narrow interpretation for numerous reasons.

86. First, multiple embodiments of the '918 patent discloses applications indirectly calling APIs. *E.g.*, EX1001, 110:12-111:17, 116:39-58; 119:49-60, FIGS. 30, 32, 35. For example, FIG. 30 depicts system 3000 for mapping traffic flow where an API (e.g., a "socket") is called by proxy 3012—a program module other than the

requesting "application"—indicating that indirect API access is contemplated by the '918 patent. EX1001, 110:12-111:17, FIGS. 30, 35.

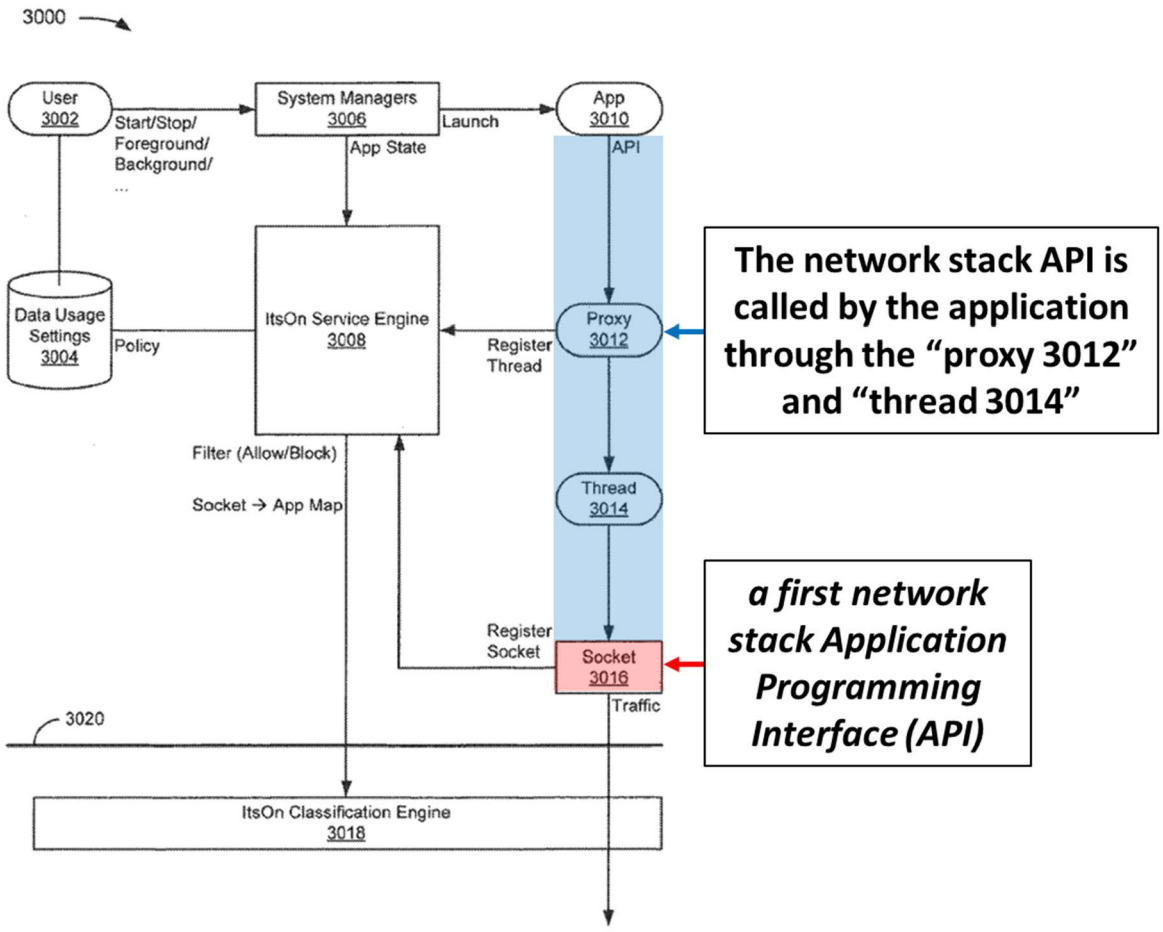


FIG. 30

EX1001, FIG. 30.

87. FIG. 32 similarly depicts system 3200 for classification mapping using virtual tagging, which includes application 3202 accessing an API (e.g., proxy service manager 3206) either directly or indirectly via control application 3204. *Id.*, 116:39-118:50, FIG. 32. FIG. 35 separately depicts system 3500 for proxy client counting in which a stack API 3518 is called via a proxy/library API 3504. *Id.*,

119:49-60, FIG. 35. These disclosed APIs remain accessible to applications in the '918 patent despite being accessed via an intermediary (e.g., control application 3204 and proxy/library API 3504).

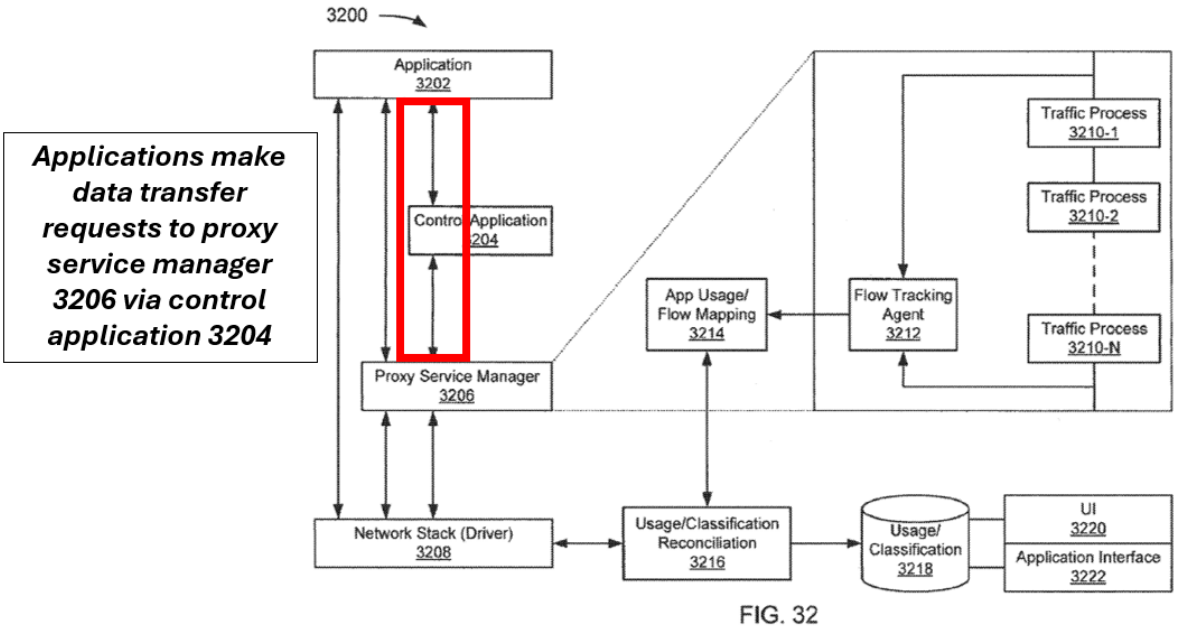


FIG. 32

EX1001, FIG. 32.

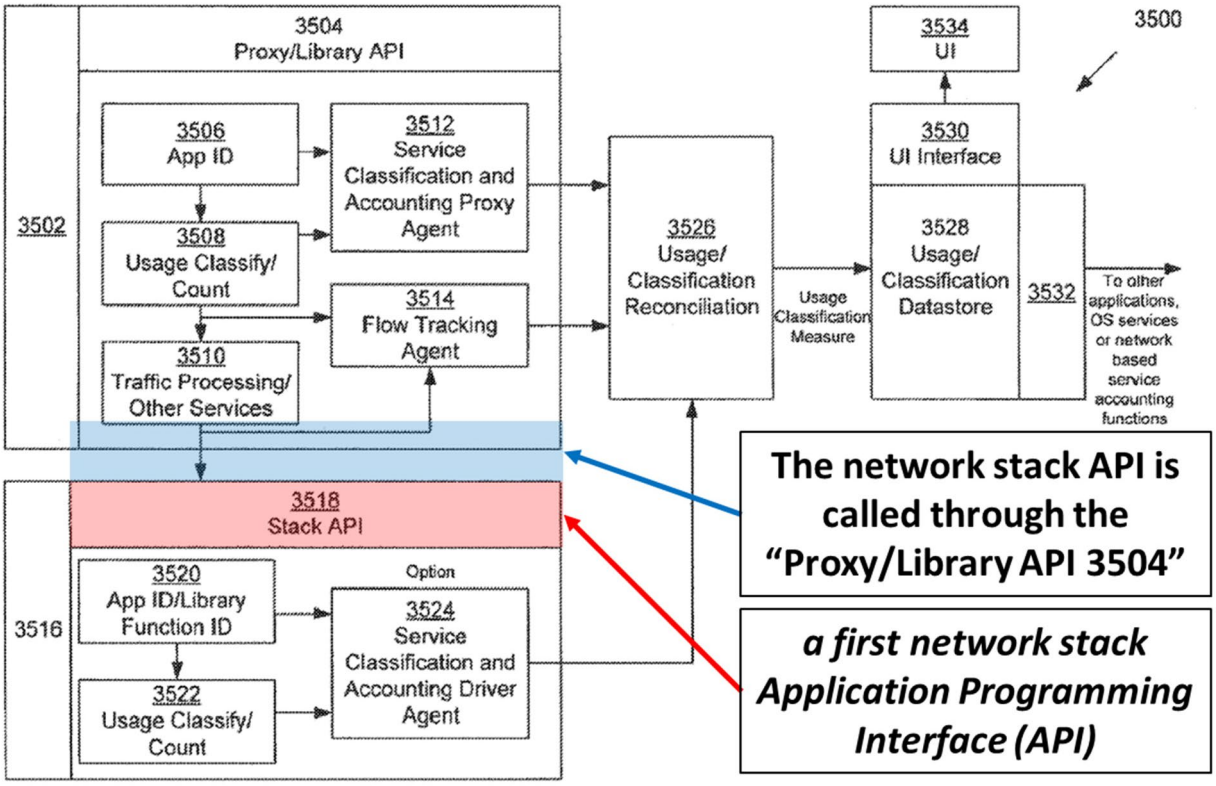


FIG. 35

EX1001, FIG. 35.

88. FIGS. 28 and 29 additionally disclose "flow trafficking" and "flow tracking" processes that enable "access" to content that is indirect. *See id.*, 105:41-110:11, FIGS. 28-29. A "content datastore 2804 … ***is accessed*** by [a] service policy implemented device 2808." *Id.*, 106:20-38, FIG. 28. In doing so, a "control application 2816 … invokes [a] proxy 2818 ***on behalf of*** the initiator 2814." *Id.*, 107:29-32, 107:60-62 ("An initiator … triggers a control application … to utilize a proxy."). These embodiments contemplate using hooks to tag network traffic, including creating a "socket-***to***-control app-***to***-initiating app" mapping of network

traffic to identify the initiating application. *Id.*, 109:47-49. The '918 patent explains that "an initiator could at least in theory have ***any number of intervening apps*** between the initiator and the proxy." EX1001, 109:26-29.

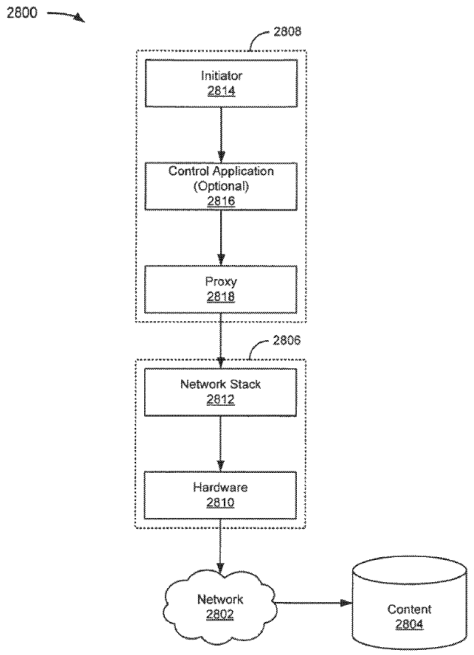


FIG. 28

EX1001, FIG. 28.

89. In still other embodiments, FIGS. 12 and 13 depict a device communication stack with an "application service interface layer" that is "above the standard networking stack API"—positioned between the API and the requesting applications (much like *Bennett*'s UA 202). EX1001, 60:61-65, 62:10-51, FIG. 12-13; EX1041, ¶25, FIG. 3.

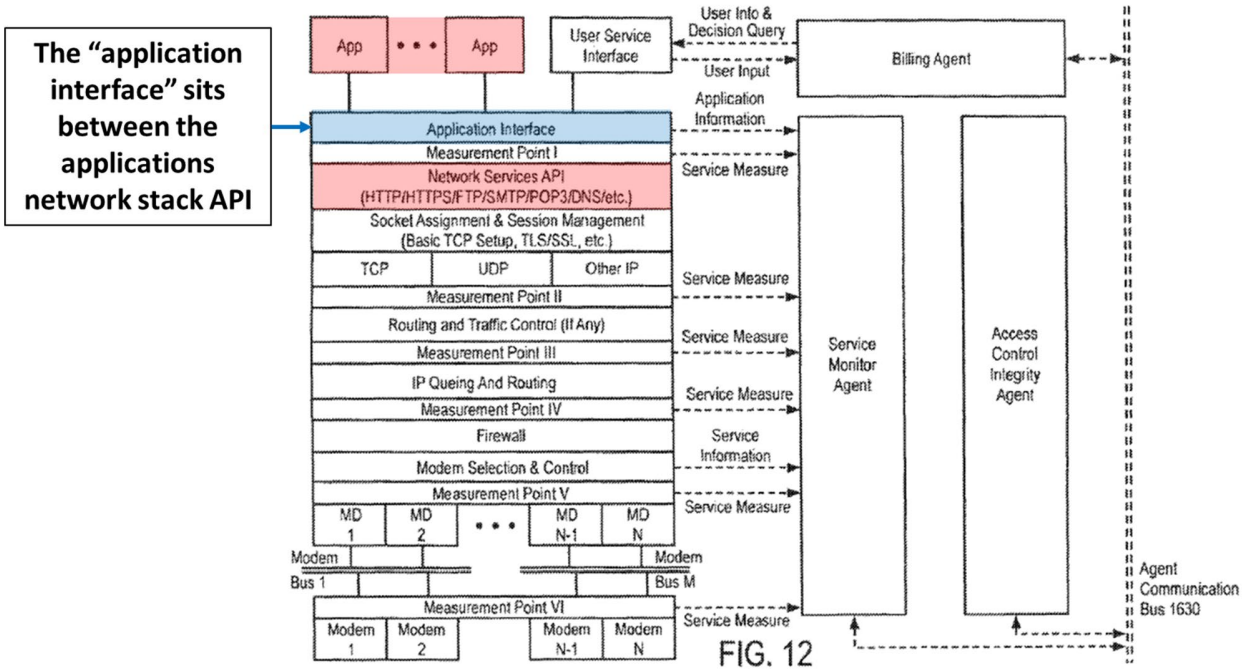


FIG. 12

EX1001, FIG. 12.

90. Yet another embodiment with indirect access—FIG. 36 —explicitly references "an API call." *Id.*, 120:7-56. In this embodiment, an "activity manager 3612" tags application traffic flow at a point that "corresponds to ***a control application call made on behalf of the app***." EX1001, 120:42-45. The activity manager 3612 "pushes the app up one more level, such that the data flow is from ***the app to a control application and to the proxy***." *Id.*, 120:45-52.

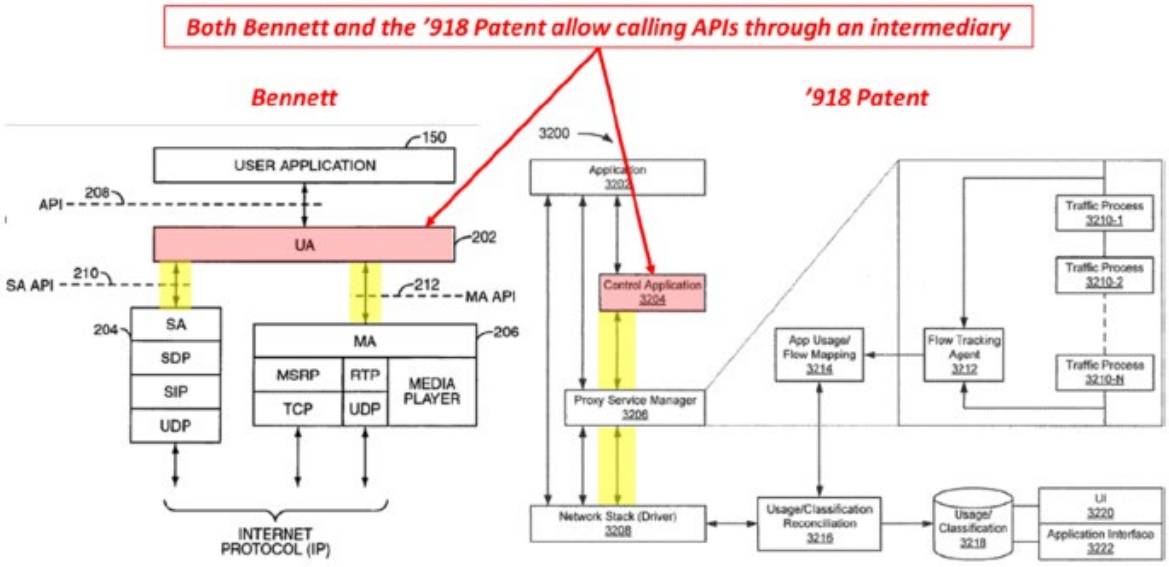
91. These embodiments confirm the plain meaning of the claims: that "access" to APIs in the claims may be direct or indirect via an intermediary. Indeed, these embodiments are consistent with a POSITA's understanding of API access as of the Critical Date, where it was known and understood that network applications

may indirectly access APIs including network stack APIs. For example, web applications running on a server do not typically directly access network stack APIs; rather, they interact with the network through "daemons" which are separate processes designed to open a socket using the socket API and wait for incoming connections. *Id*. When a request arrives on the socket, the request is passed to the web server application for processing; but the socket API is still clearly "accessed" via the daemon by the web application. *Id*. Like the '918 patent, the prior art similarly uses the term "accessible" broadly. *e.g.*, EX1042, ¶13 (apps communicating with other apps that are "accessible via a network"); EX1043, 2:35-40 (remote servers are "accessible" to content publishers); EX1054, 20:58-21:3 (operating system calls gather "network accessible information"). These examples illustrate the plain and ordinary meaning of "accessible" as including both "direct" and "indirect" accessibility. *Id*.

92. Here, the claims do not recite the term "directly" nor do they otherwise preclude "indirect" access. Nor did Headwater seek that construction previously in the district court, conceding this feature should be interpreted according to its plain and ordinary meaning. *See generally* EX1060.

93. *Bennett* discloses APIs that are "accessible" to its applications, consistent with the proper scope of the claims. Indeed, *Bennett* closely aligns with the '918 patent's embodiments: *Bennett*'s UA 202 is interposed between its

application and API just like the '918 patent discloses with respect to control application 3204:



Compare EX1041, FIG. 3 *and* EX1001, FIG. 32.

e. 1[d]: *"a second API containing at least one second call accessible to each of the plurality of device applications, the second API callable by each of the plurality of device applications to make a data transfer request for a media object associated with a network resource identifier supplied by the calling device application"*

"a data transfer request for a media object"

94. The '918 patent describes "media download[s]," "media streaming," "audio files" played by a "media player," "streaming audio," "video conference[s]," Voice over Internet Protocol ("VoIP"), "multimedia data," and "instant messaging" as example application activities that involve the transfer of media ("***a data transfer request for a media object***"). EX1001, 71:31-42, 72:37-50, 107:34-46, 111:44-

112:17. Consistent with the '918 patent, *Bennett* discloses similar "***media objects***" retrieved after a "***data transfer request***," for example, "voice over IP (VoIP), video and audio streaming, … videoconferencing, [and] instant messaging." EX1041, ¶¶17-18, 24, 70, 76.

"a second API containing at least one second call accessible to each of the plurality of device applications, the second API callable by each of the plurality of device applications"

95. *Bennett* discloses that its media client 200 includes a "media agent (MA …) 206" that "manages media connections, routes media according to media type and user settings, and invokes media players to process media as required" ("***make a data transfer request for a media object***"). EX1041, ¶¶25, 50-56, Table-3, FIGS. 3-10. The MA 206 is called by the UA 202 of the media client 200 using a "MA API 212" ("***a second API***") in response to a request from a "user application 150" ("***the second API callable by each of the plurality of device applications***"). EX1041, ¶¶31, 50-56, Table-3, FIGS. 3-10; *see supra* §X.A.4.d. (1[c]) (describing a "***plurality of device applications***").

96. The MA API 212 includes various "requests" to send and receive media, including "LISTEN," "SEND," and "OPEN" requests ("make a data transfer request for a media object"). EX1041, ¶¶31, 33, 40, 45-46, FIGS. 6-7, Table-2; *see supra* §X.A.4.d. (1[c]).

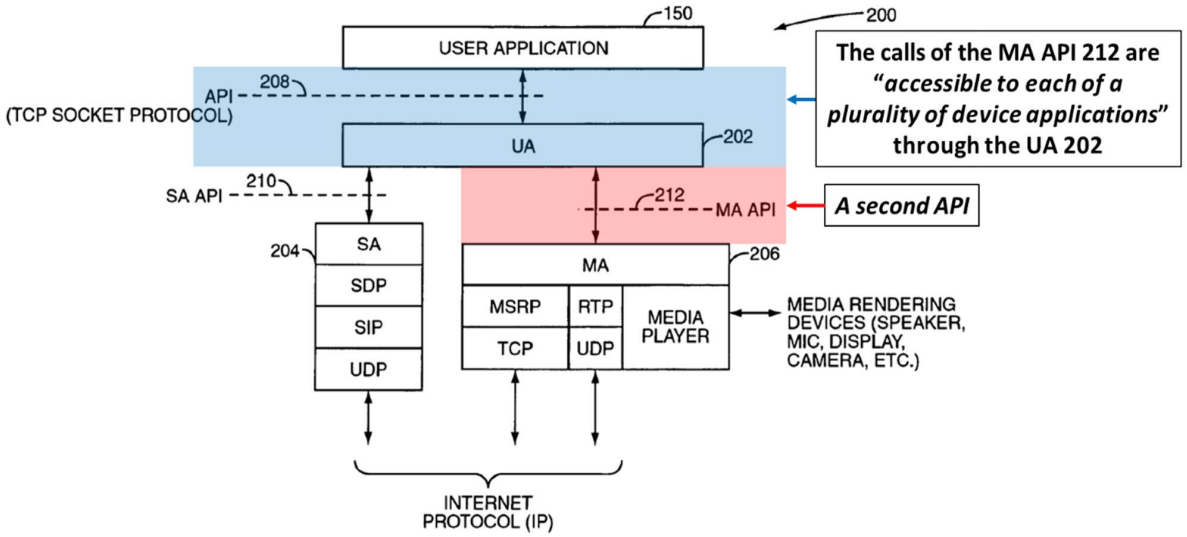


FIG. 3

EX1041, FIG. 3.

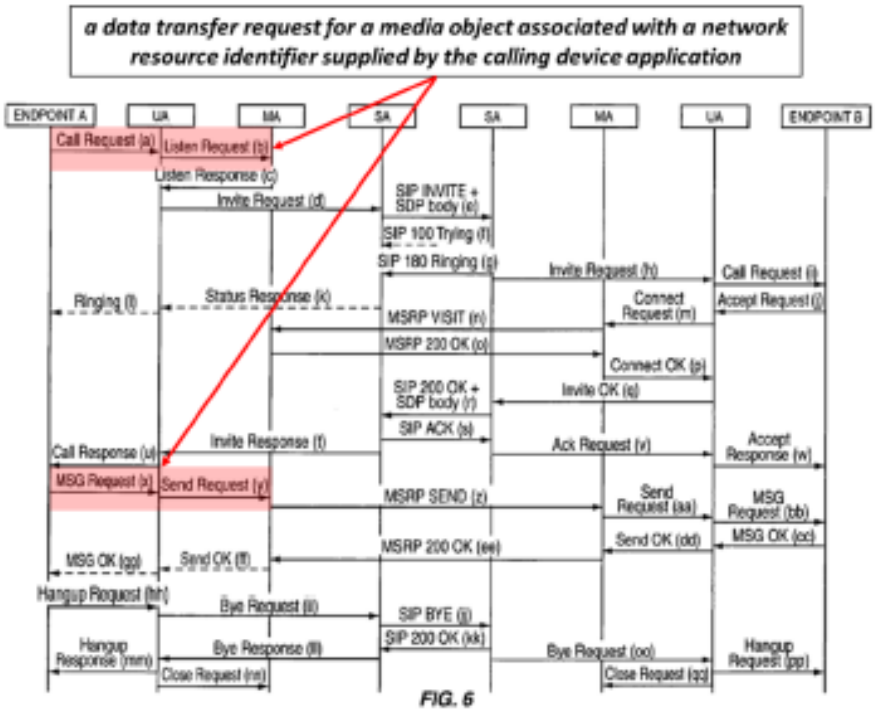


FIG. 6

EX1041, FIG. 6.

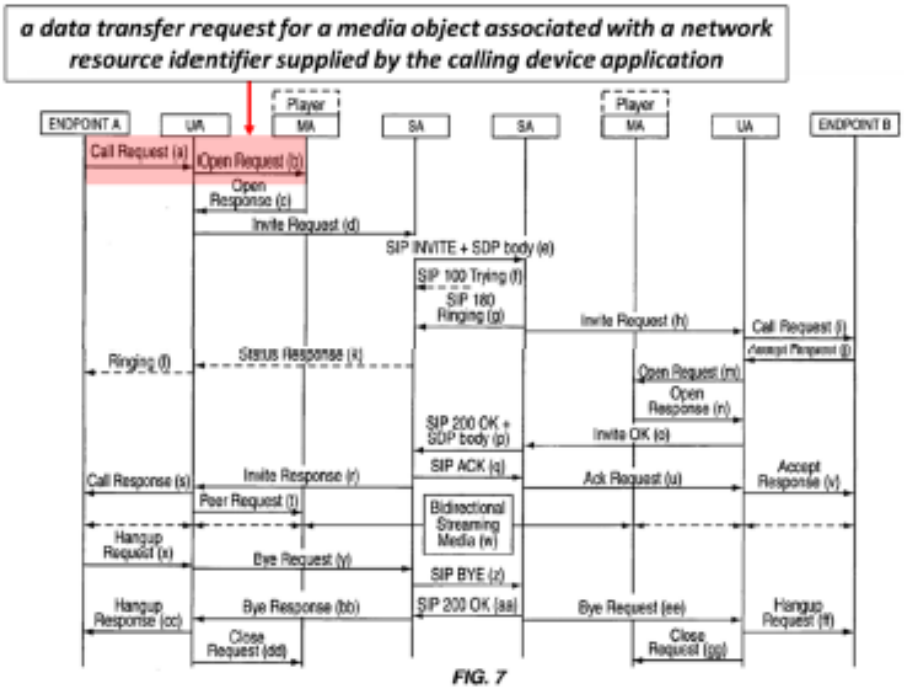


FIG. 7

EX1041, FIG. 7.

97. To the extent there is a question that the calls of the "***second API***" must be directly "***accessible to each of a plurality of device applications***," or that the application must directly call the "***second***" API, the '918 patent does not support such a narrow interpretation. *See supra* §X.A.4.e. (1[d]). The process of applications indirectly calling APIs is instead disclosed in the '918 patent in multiple embodiments. *See supra* §X.A.4.e. (1[d]).

"data transfer request for a media object associated with a network resource identifier supplied by the calling device application"

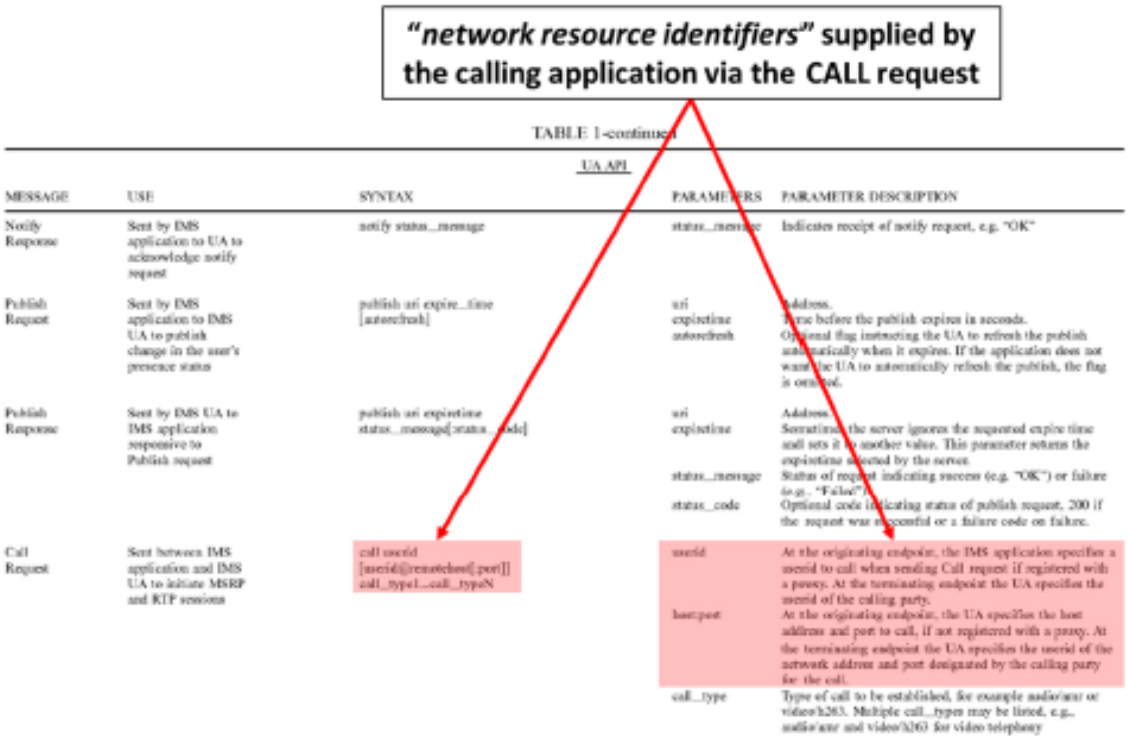
98. The '918 patent gives examples of "***network resource identifier[s]***" that include: "(e.g., an IP address, a URL, a remote file name or address, a stream name, an object name, or any combination of these identifiers) that identifies a

source (or a proxy to the source) of the data to be transferred or a data object to be transferred." EX1001, 112:37-47.

99. In the context of a media request, such as a request to initiate a voice over IP or audio connection, *Bennett* discloses that its MA API 212 calls, for example the OPEN request, which includes "a network address of a remote host from which media connections will be accepted" ("***data transfer request for a media object associated with a network resource identifier supplied by the calling device application***") that is provided by the application when it calls the UA 202 (e.g., via a "CALL" request) ("***supplied by the calling device application***"). EX1041, ¶¶34, 50-56, 79 ("the application 150 … initiates the media session by sending a CALL request to the UA 202, … [which] sends an OPEN request to the MA 206 in the remote video player 350 to open a UDP socket connection for the RTP session."), 86, Table-3. Applications also provide a "userid" and a "port" to call via the CALL request ("***a network resource identifier***"). EX1041, ¶84, Table-1.

100. *Bennett*'s examples of "***network resource identifier[s]***" are consistent with the '918 patent, and include: "a userID, alias, or fully qualified network address" ("an IP address" and "remote file … address"). EX1041, ¶34; EX1001, 112:42-45; *see also* EX1041, Table-1, Table-3. *Bennett* discloses that the CALL request includes this information, including "identifying the called party and the call type, such as a user ID, alias, or fully qualified network address." *Id.*, ¶34. *Bennett*

also provides example addresses of "call alice@ims.net: 5060 video/h263" and "call 10.0.0.1:5060 video/h263" (a "remote file … address"), as shown in the Tables below. *Id.*; *see also id.*, Table-1, Table-3. Additionally, all the aforementioned examples "identif[y] a source (or a proxy to the source) of the data to be transferred or a data object to be transferred." EX1041, ¶34; EX1001, 112:43-47. *Bennett* also describes media located at Uniform Resource Identifiers ("URIs"), which a POSITA would have recognized include Uniform Resource Locators (URLs). EX1041, Table-1.



EX1041, Table-1.

The OPEN request of the MA API includes the "*network resource identifier*" supplied by the calling application

TABLE 3

MA API

MESSAGE	USE	SYNTAX	PARAMETERS	PARAMETER DESCRIPTION
Listen Request	Sent by UA to MA to initiate a MSRP session. The MA opens a TCP listener in response to the Listen request.	listen [remotehost]	remotehost	Optional parameter specifies address from which connections can be made.
Listen Response	Sent by MA to UA as final response to Listen request. The Listen response includes the address and port of the TCP connection opened for the MSRP session.	listen status_message[:status_code] host:port	status_message status_code host:port	Status of listen request indicating success (e.g. "OK") or failure (e.g., "Failed") Optional code indicating status of Listen request, 200 if the request was successful or an error code on failure. Network address of host and port number for port opened in response to Listen Request. Returned when Listen request is successful. Omitted when Listen request fails.
Open Request	Sent by UA to MA to initiate RTP session. The MA opens a TCP connection in response to the Open request.	open [remotehost]	remotehost	Optional address specifies address from which connections can be made.

EX1041, Table-3.

f. 1[e]: *"a media service manager prompted by the second call, to manage network data transfers for the media object by interfacing with the network stack to retrieve the media object associated with the network resource identifier via the wireless modem and the wireless network"*

101. *Bennett* discloses that its media client 200 includes a "media agent (MA …) 206" ("***a media service manager***") that "implements the message session relay protocol (MSRP) and the Real-Time Transport Protocol (RTP)" and "manages media connections, routes media according to media type and user settings, and invokes media players to process media as required" ("***manage network data transfers for the media object by interfacing with the network stack to retrieve the media object***"). EX1041, ¶¶25, 50-56, Table-3, FIGS. 3-10. *Bennett* discloses several MA API requests that disclose "***transfer[ring] for the media object***" during

a voice over IP, video and audio streaming, for example sending and receiving media via "LISTEN," "SEND," and "OPEN" requests. EX1041, ¶¶31, 33, 40, 45-46, 50, 54, FIGS. 6-7, Tables 2-3; *see supra* §X.A.4.e. (1[d]).

102. The MA 206 is called by the UA 202 of the media client 200 using a "MA API 212" ("***prompted by the second call***") in response to a request from a "user application 150." EX1041, ¶¶31, 50-56, Table-3, FIGS. 3-10. Additionally, the MA 206 uses "TCP and/or UDP over IP for transport of RTP and MSRP messages" and retrieves media that "passes up through the IP, UDP and RTP stacks" prior to being played at a media player or decoder ("***interfacing with the network stack to retrieve the media object***"). EX1041, ¶¶25, 76-79. Additionally, as described above, *Bennett*'s media is retrieved from "remote host[s]" over wireless networks (e.g., "GPRS") through the use of a corresponding wireless modem (such that *Bennett*'s media is retrieved "***via the wireless modem and the wireless network***"). EX1041, ¶¶17, 54, FIG. 1; *see supra* §X.A.4.b. (1[a]).

103. The media object retrieved in *Bennett* is associated with the network resource identifier. As described above, requests from applications include a "***network resource identifier***" that is "***associated***" with the "***media object***." *See supra* §X.A.4.e. (1[d]). The MA 206 responds with media that is based on the request from the user application, and thus retrieves a media object associated with a network resource identifier supplied by the calling device application. For example,

LISTEN requests connections "will be accepted only from the specified host" and "[i]n response to the LISTEN request, the MA 206 opens a port for a media connection and sends a LISTEN response to the UA 202 giving the address and port for the media connection." EX1041, ¶51. In response to a SEND request to send multimedia via an MSRP session, the MA 206 "sends the message" and "[w]hen the message is acknowledged, the MA 206 sends a SEND response to the UA 202 identifying the call and indicating the status of the SEND request." *Id.*, ¶53. For OPEN requests, "media connections will be accepted only from the address specified in the OPEN request." *Id.*, ¶54. The MA 206 "opens a port for a media connection and returns an OPEN response indicating the network address and port for the media connection." *Id.*

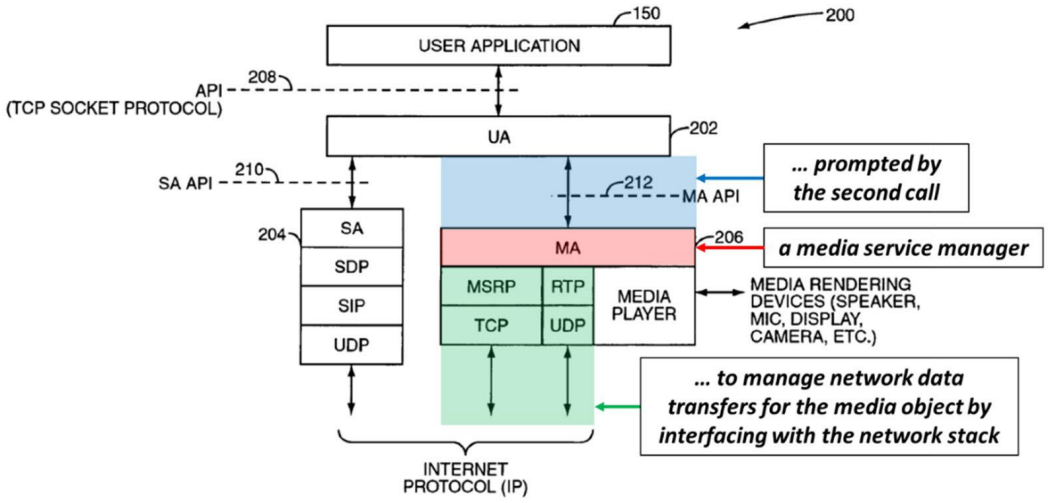
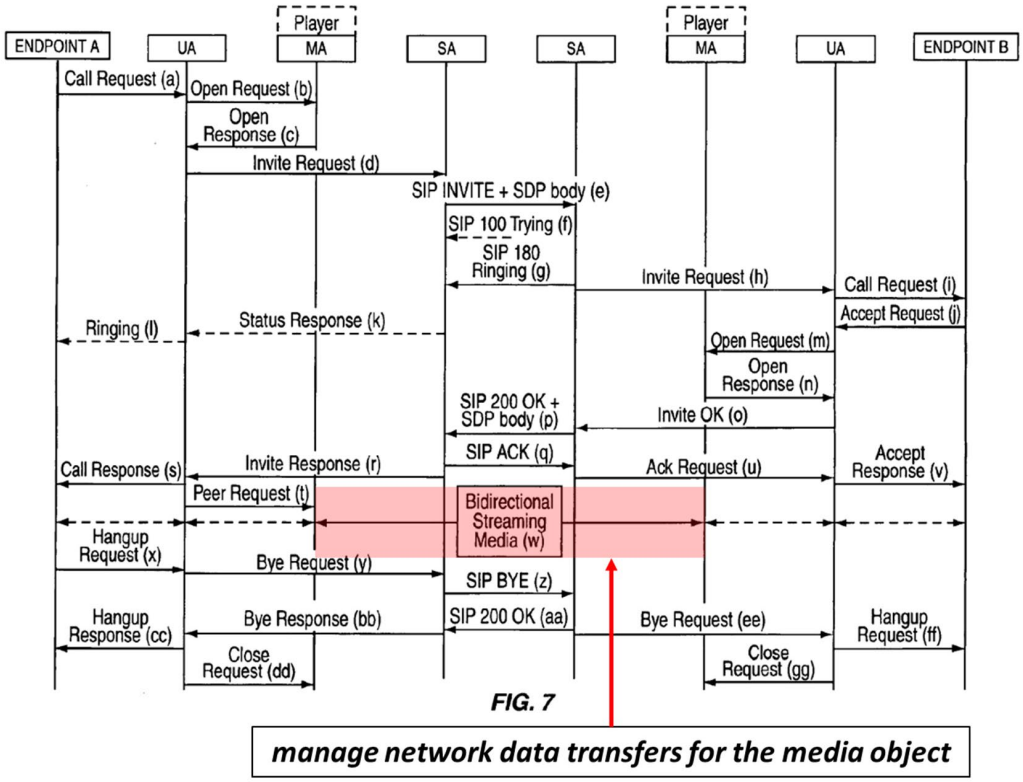


FIG. 3

EX1041, FIG. 3.



EX1041, FIG. 7.

g. 1[f]: *"one or more service classification and measurement agents to associate wireless network data usage for the media object network data transfers with the device application that requests the data transfer for the media object, to associate wireless network data usage for respective data packet flows opened and used via the first network stack API with the device application opening such respective data packet flow, and to reconcile wireless network data usage for each of the plurality of device applications to track an aggregate wireless network data usage attributable to each of the plurality of device applications via both the first network stack API and the second API"*

104. A service classification and measurement agent in 1[f] associates wireless network data usage via the first network stack API and the second API. *Vadde* discloses these features in the *Bennett-Vadde* combination. For example, *Vadde* discloses techniques for "managing data traffic" for each of a plurality of applications using a "policy" based system that enforces restrictions based on "attributes" in the transmitted data and corresponding data "usage limits." EX1042, ¶¶10, 15-16, 25-26. *Bennett* in view of *Vadde* renders 1[f] obvious.

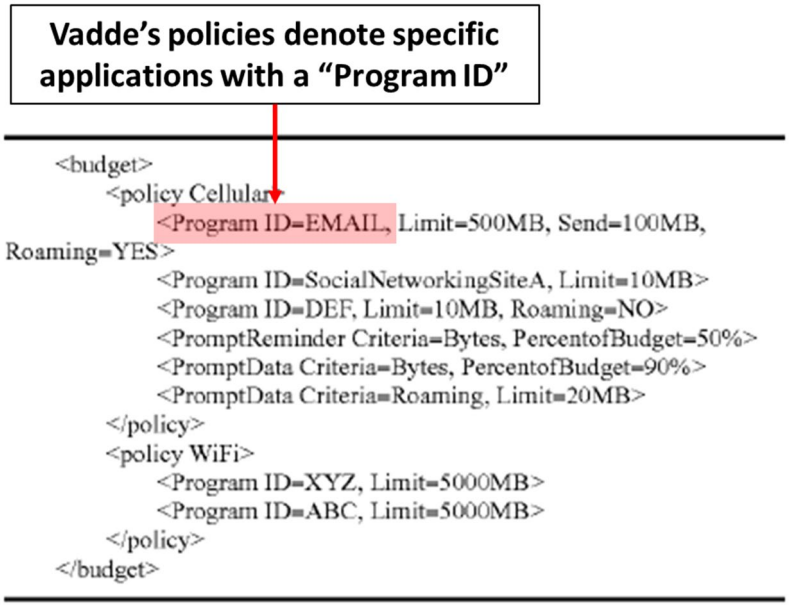
"one or more service classification and measurement agents to associate wireless network data usage for the media object network data transfers with the device application that requests the data transfer for the media object"

105. This feature associates wireless network data usage via the second API. *Vadde* enforces its policies with a "restrictor component 122" ("***one or more service classification and measurement agents***") which "appl[ies] the data usage policy" and "monitors the data transmitted and/or received by the applications 110 and

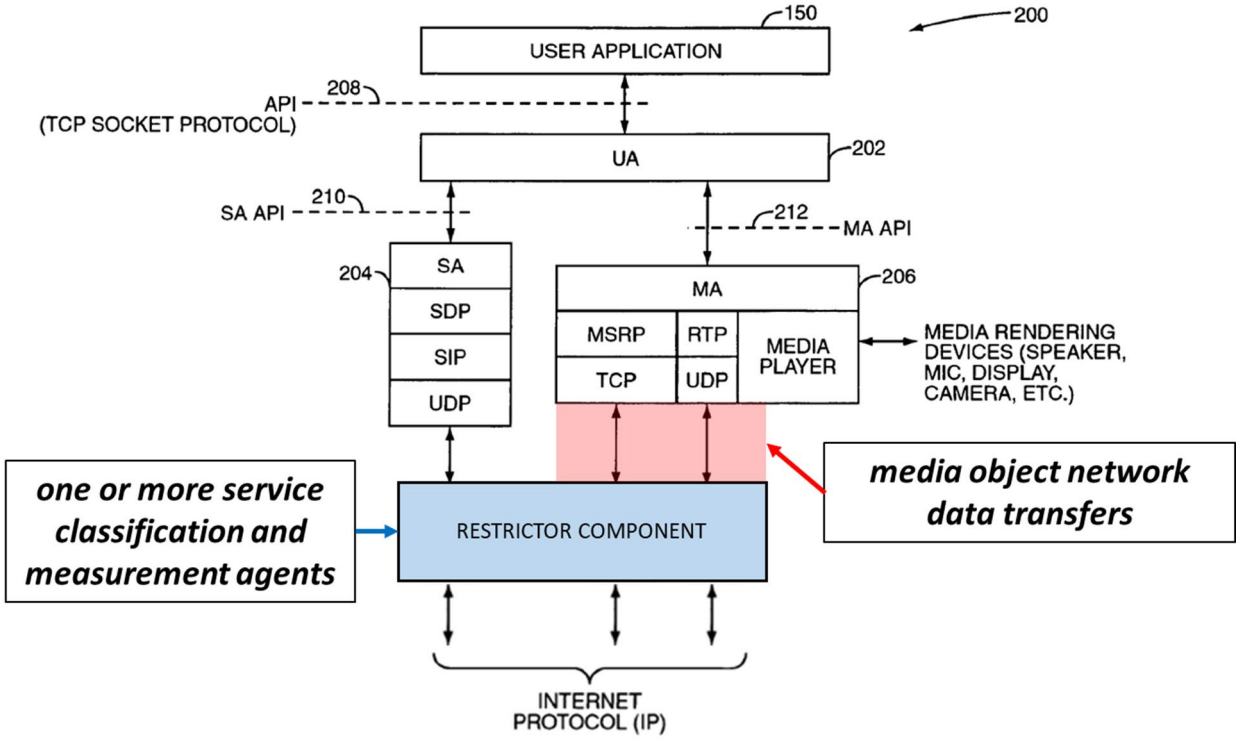
determines whether the data usage limits 116 corresponding to each of the applications 110 have been exceeded or are about to be exceeded." EX1042, ¶¶22, 24-26, 29-32. *Vadde*'s restrictor component 122 "monitors the data transmitted and/or received by the applications 110 and determines whether the data usage limits 116 corresponding to each of the applications 110 have been exceeded or are about to be exceeded (e.g., within a percentage such as 80% of the maximum or absolute threshold)" ("***service classification and measurement***"). EX1042, ¶22. This includes monitoring transmitted data to "by receiving or accessing the attributes 112 or other statistics describing the data transmitted and/or received by each of the applications 110." *Id.* ¶22. Exemplary "attributes 112" include an "application name," which is given a "Program ID." *Id.* ¶¶24, 32. *Vadde* discloses that "data traffic is monitored ***per application 110***." *Id.* ¶9. *Vadde* also discloses monitoring "a corresponding set of attribute values" to produce "usage patterns" to "to restrict data traffic per application 110." EX1042, ¶¶9, 24.

106. By monitoring data usage "corresponding" to each application, including accessing attributes that identify (e.g., classify) specific applications on a "per application" basis, and determining whether that usage exceeds a threshold (e.g., measure), restrictor component 122 discloses a "***service classification and measurement agent … to associate wireless network data usage for … the device application***."

107. In combining *Vadde*'s techniques into *Bennett*'s device, as explained above, a POSITA would have found it obvious to leverage *Vadde*'s monitoring of application data usage to monitor and associate "***media object network data transfers***" performed via the MA API 212 with "***the device application that requests the data transfer for the media object***" as these transfers would have resulted in data usage that could potentially be chargeable to the user or capped in the *Bennett-Vadde* device. *See supra* §X.A.3; *e.g.*, EX1042, ¶¶1, 34-35. As discussed above, monitoring on a per-application basis would have allowed data-intensive applications to be identified to prevent them from consuming excessive data (e.g., "restrict data traffic per application" and prevent "mobile operator charges," EX1042, ¶9) or drawing excessive device power (e.g., "reduce battery usage, … , and/or processor usage," *id.*, ¶9). *See supra* §X.A.3; *see also* EX1042, ¶¶23 (describing "provid[ing] the user 104 with a data cost associated with each of the applications 110" and "resource cost corresponding to an amount of battery power consumed."), 30 (restricting data for "the application 110 deemed to be in violation of the data usage limits 116"). It also would have enabled additional applications and functionality into wireless devices while allowing users to retain control of aggregate device data usage. EX1042, ¶¶10, 16, 18, 22, 25, 29-30, 32.



EX1042, ¶32.



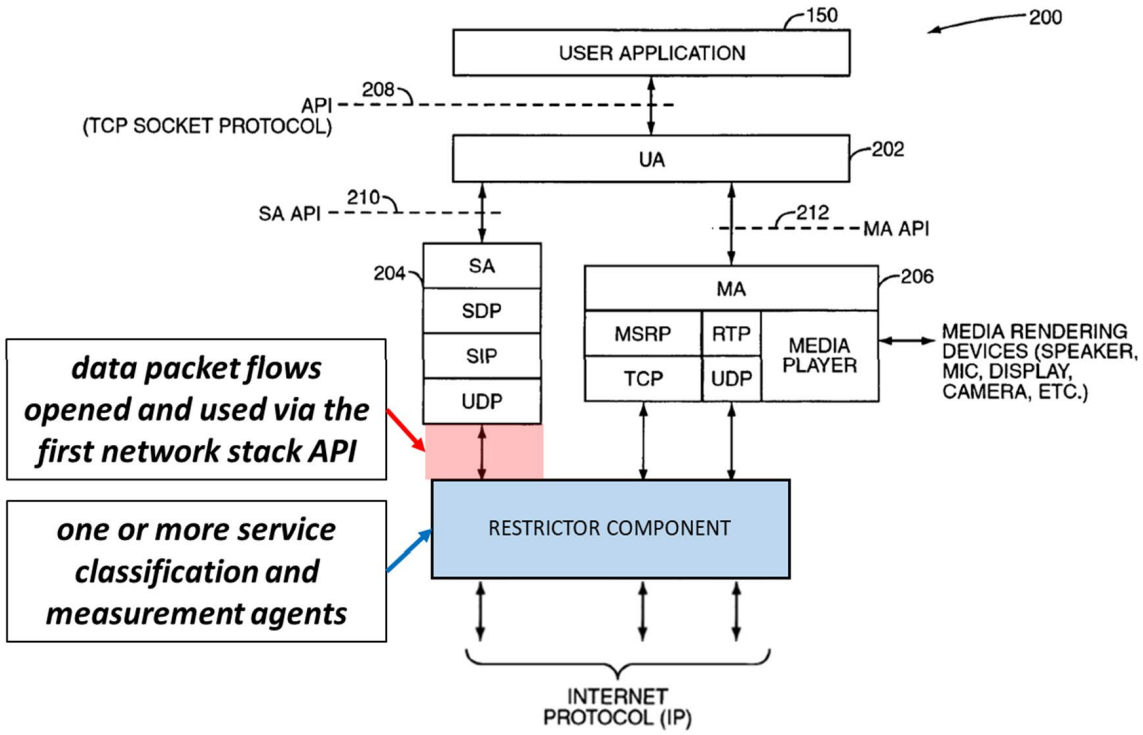
EX1041, FIG. 3 (as modified by *Vadde*).

"to associate wireless network data usage for respective data packet flows opened and used via the first network stack API with the device application opening such respective data packet flow"

108. This feature associates applications with wireless network data usage via the first network stack API. As explained above, a POSITA would have found it obvious to leverage *Vadde*'s monitoring of application data usage to monitor and associate "***respective data packet flows opened and used***" via the SA API 210 ("***the first network stack API***") with "***the device application opening such respective data packet flow***" as these packet flows would have resulted in data usage that would potentially be chargeable to the user of the *Bennett-Vadde* device. EX1042, ¶¶10, 15-16, 25-26; *see supra* §X.A.3. As discussed, associating data usage on a per-application basis would identify data-intensive applications and those that draw excessive device power, as well as enable additional applications to be added while maintaining control of data usage. *See supra* §X.A.3; EX1042 ¶¶9, 10, 16, 22, 23, 25, 29-30, 32.

109. A POSITA would have recognized and found obvious that the protocols disclosed in *Bennett* would have included "***data packet flows***" as these protocols are examples of "packet switched services" that communicate data in a series of data packets. EX1041, ¶17; *see supra* §X.A.4.d (1[c]). For example, *Rakoshitz* describes that network traffic is "a ***flow of information*** or data or packets of information." EX1046, 12:12-58, 15:57-67.

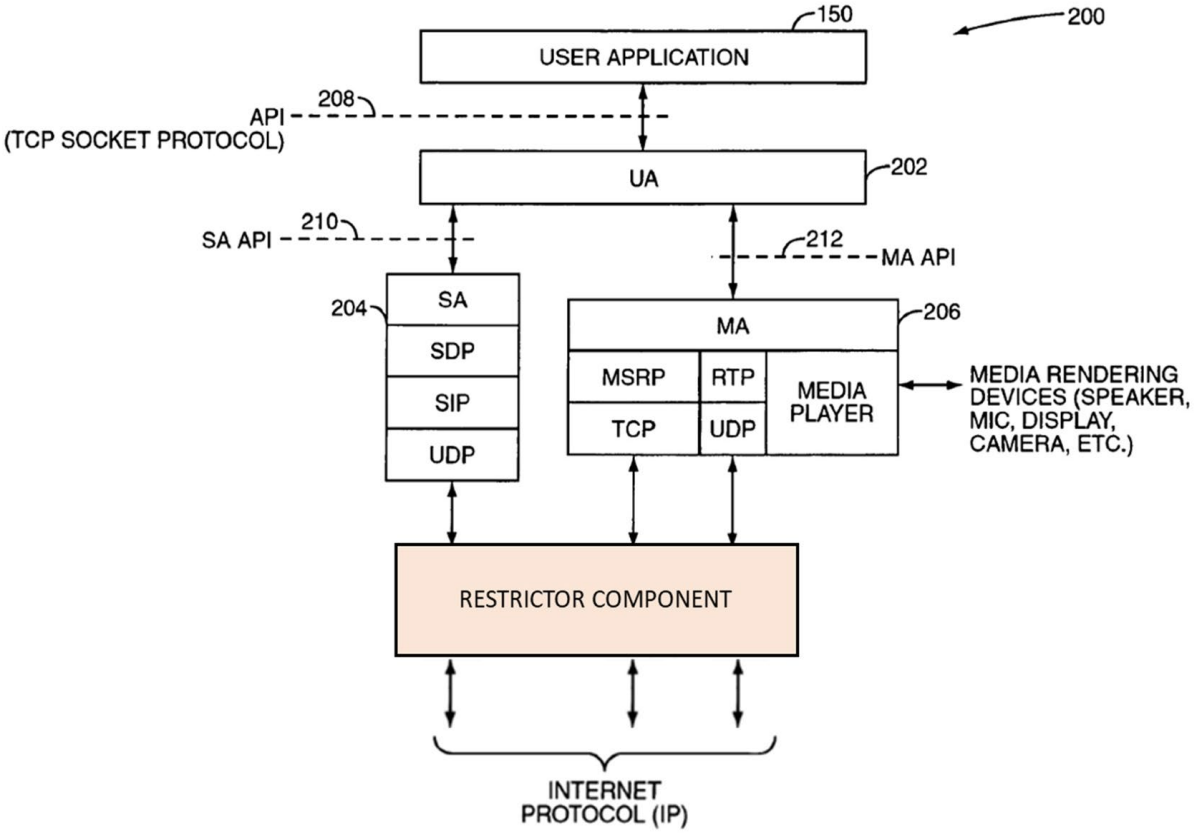
110. In the combination, the restrictor component of the *Bennett-Vadde* device would have monitored "[d]ata transmitted and/or received by each of a plurality of applications executing on a computing device" on a "per application" basis. EX1042, ¶¶2, 9, 15-16, 22. By monitoring and measuring data transmitted by applications, the restrictor component measures data usage associated with the "flow" of information from those applications (communication sessions established via the SA 204 and SA API 210—"***respective data packet flows opened and used via the first network stack API***") classified to each application ("***with the device application opening such respective data packet flow***"). EX1042, ¶¶10, 15-16, 25-26.



EX1041, FIG. 3 (as modified by *Vadde*).

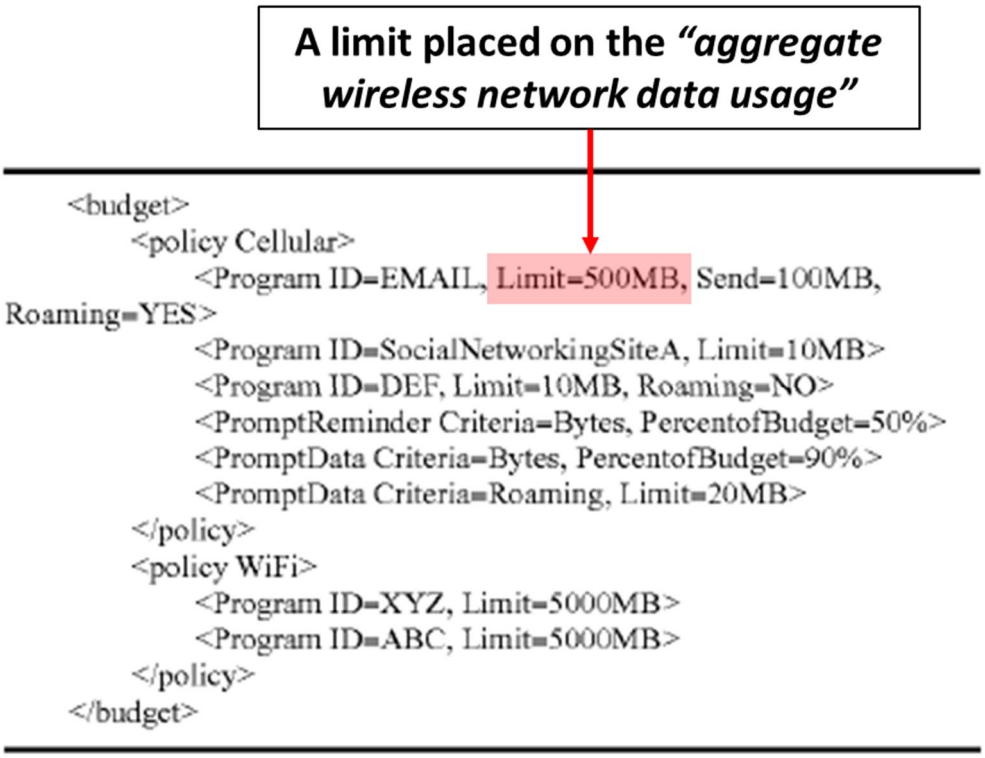
"and to reconcile wireless network data usage for each of the plurality of device applications to track an aggregate wireless network data usage attributable to each of the plurality of device applications via both the first network stack API and the second API"

111. *Vadde*'s restrictor component 122 "monitors the data transmitted and/or received by the applications 110 and determines whether the data usage limits 116 corresponding to each of the applications 110 have been exceeded or are about to be exceeded." EX1042, ¶¶22, 24-26, 29-32. In the combined device, as discussed above, *Vadde*'s restrictor component 122 would monitor application data usage via the SA API 210 and the MA API 212 and attribute that usage to the application, as *Vadde* contemplates. EX1042, ¶¶2 ("manag[ing] data traffic per application"), 23 ("provid[ing] the user 104 with a data cost associated with each of the applications 110."); *see supra* §X.A.3. By monitoring data traffic from applications through both APIs in the combined system and determining whether data usage limits corresponding to each application have been exceeded, *Bennett* in view of *Vadde* renders obvious "***reconcil[ing] wireless network data usage for each of the plurality of device applications***."



EX1042, FIG. 3 (as modified by *Vadde*).

112. *Vadde*'s "data usage limits 116" are set for "each" application and include "a maximum quantity of data to be transmitted and/or received by a particular application 110" ("***an aggregate wireless network data usage attributable to each of the plurality of device applications***"). EX1042, ¶¶10, 16, 18, 22, 25, 29-30, 32. As described above, this data usage would have been monitored for both the SA API 210 ("***the first network stack API***") and MA API 212 ("***the second API***") in the *Bennett-Vadde* combination. *See supra* §X.A.3.

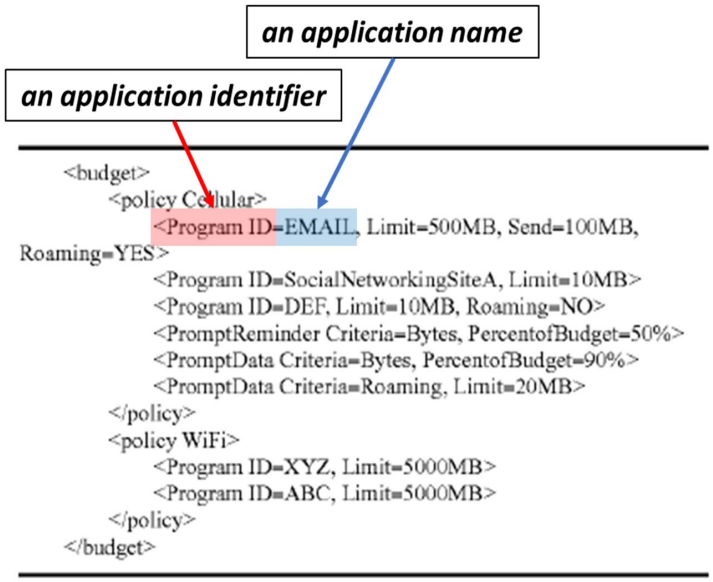


EX1042, ¶32.

5. Claim 2: "*The wireless end-user device of claim 1, wherein to associate wireless network data usage for the media object network data transfers with the device application that makes the data transfer request for the media object comprises to identify at least one of an application name, an application identifier, or a process identifier for the application that makes the data transfer request.*"

113. Vadde discloses monitoring data traffic and "defin[ing] … one or more attributes" in the data, including application name. EX1042, ¶24. *Vadde* discloses that each application's usage is tracked using attributes that include an "application name" ("***an application name***") that is a "Program ID" ("***an application identifier***"). EX1042, ¶¶24, 32; *see supra* §X.A.4.g (1[f]). By defining attributes in the data

traffic that include an application name and Program ID, *Vadde* discloses "***associat[ing] wireless network data usage for the media object network data transfers with the device application***" by "***identify[ing] … an application name, [and] an application identifier, … for the application that makes the data transfer request***." *Id*. The *Bennett-Vadde* device therefore renders claim 2 obvious.

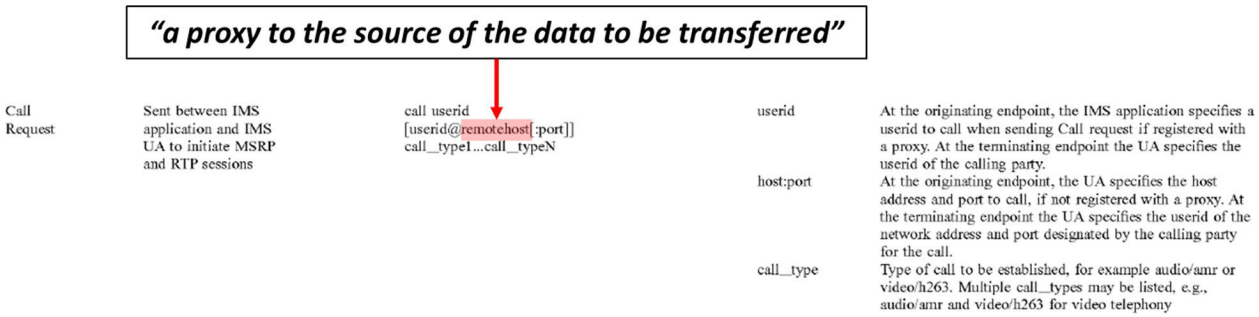


EX1042, ¶32.

6. Claim 3: *"The wireless end-user device of claim 2, wherein the data transfer request comprises a network resource identifier that identifies a source of the data to be transferred, a proxy to the source of the data to be transferred, or the media object to be transferred, in particular, wherein the network resource identifier comprises one or more of an Internet Protocol address, a Uniform Resource Locator, a remote file name/address, a stream name, and an object name."*

114. As discussed above, *Bennett*'s "requests" ("***the data transfer request***") include "a network address of a remote host from which media connections will be

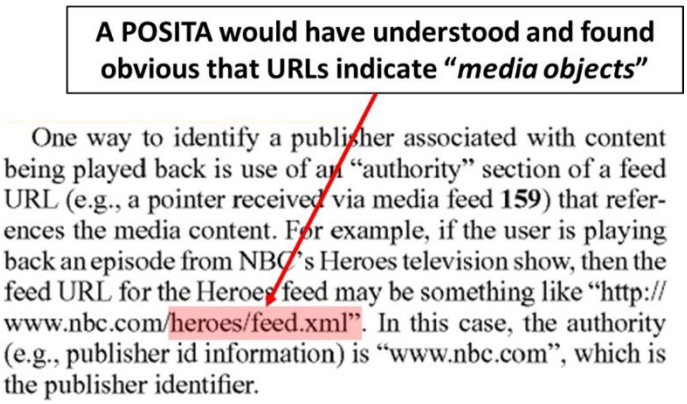
accepted" ("***a network resource identifier that identifies a source of the data to be transferred***") that is provided by the application when it calls the UA 202 (e.g., via a "CALL" request). EX1041, ¶¶34, 50-56, Table-3; *see supra* §X.A.4.e (1[d]). Additionally, the CALL request includes a "host address and port to call ("@remotehost"—"***a proxy to the source of the data to be transferred***") when the recipient has not previously registered with the proxy. EX1041, ¶¶20, 34, 84, Table-1.



EX1041, Table-1.

115. *Bennett*'s examples of "***network resource identifier[s]***" include: "a userID, alias, or fully qualified network address" ("***a remote file name/address***"). EX1041, ¶34. *Bennett* also provides example addresses of "call alice@ims.net: 5060 video/h263" and "call 10.0.0.1:5060 video/h263" ("***a remote file name/address***"). *Id*. *Bennett* further describes content located at Uniform Resource Identifiers ("URIs"), which a POSITA would have recognized include "***Uniform Resource Locator[s]***" ("URLs"). EX1041, Table-1; *see supra* §X.A.4.e (1[d]). Further, in my opinion, a POSITA also would have recognized and found obvious that URLs also

indicated "***the media object to be transferred, in particular***" as the file name of the media object would typically comprise part of the URL. For example, *Riggs* discloses that "the global address of content" ("***the media object to be transferred, in particular***") is "typically" provided "in the form of a Uniform Resource Locator (URL)" (e.g., a "feed URL"). EX1043, 1:25-35, 5:54-62, 6:38-43.

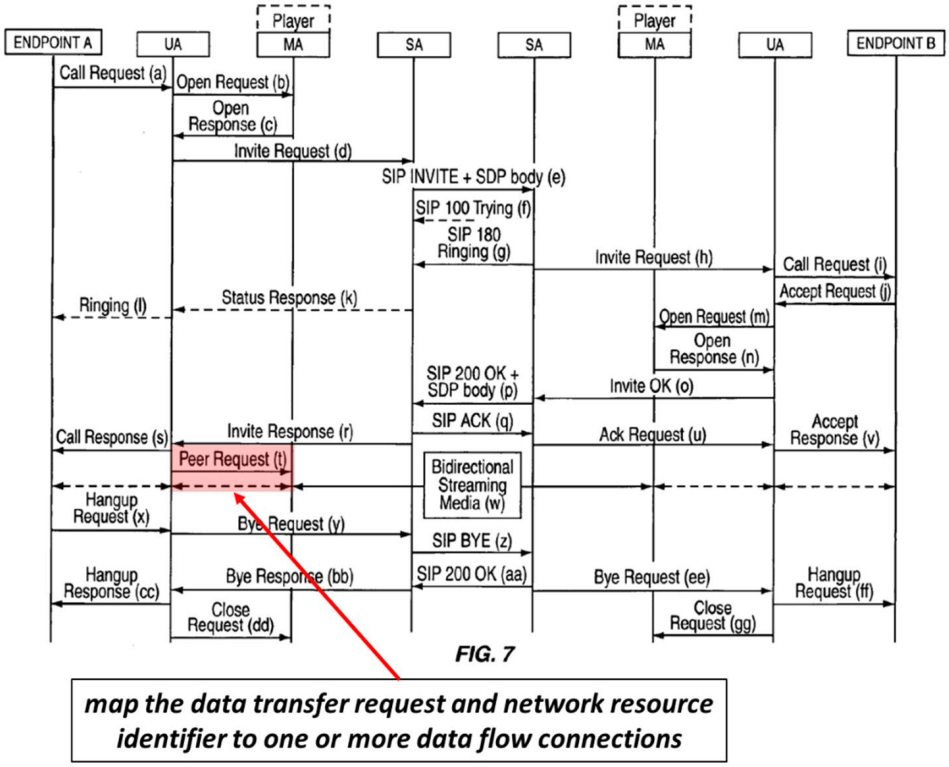


EX1043, 5:54-62.

7. Claim 8: *"The wireless end-user device of claim 3, wherein to manage network data transfers for the media object by interfacing with the at least one network stack comprises to map the data transfer request and network resource identifier to one or more data flow connections communicated through the device network stack."*

116. *Bennett* discloses that, once the SA 204 establishes a communication session (e.g., an "RTP session"—"***one or more data flow connections communicated through the device network stack***"), the UA 202 "sends a PEER request to the MA206 to provide the MA206 with the host address and port opened for the RTP session" ("***map the data transfer request and network resource identifier to one or more data flow connections communicated through the device***

network stack"). EX1041, ¶¶55, 73, 79, FIG. 7. The PEER request includes "the network address and port for the media connection" (the "***network resource identifier***" associated with the "***one or more data flow connections communicated through the device network stack***"). *Id*.



EX1041, FIG. 7.

8. Claim 9: *"The wireless end-user device of claim 1, further comprising a media player and a user interface, wherein the media object comprises media data that is, as a result of the media service manager management of network data transfers for the media object, received by the device and played by the media player through the user interface."*

117. *Bennett* discloses that its media client 200 includes "one or more media players" ("***a media player***") to "process and output media to media rendering

devices" which include a "speaker and/or display of a mobile terminal 100." EX1041, ¶¶2, 25, 76, FIGS. 3, 9, 11. As an example, *Bennett* discloses "media streaming" (a "***media object [that] comprises media data***") where "the user application 150 … receives the media stream and outputs the media stream to a media player" and, in this situation, the "MA 206 … directly route[s] the media stream to a media player" ("***that is, as a result of the media service manager management of network data transfers for the media object, received by the device and played by the media player through the user interface***"). EX1041, ¶76, FIG. 9. This may occur during, for example, a videoconferencing session following a CALL request. *See supra* §X.A.4.f (1[e]); EX1041, ¶¶17, 18, 24. *Bennett*'s FIGS. 3 and 9 depict examples of this process.

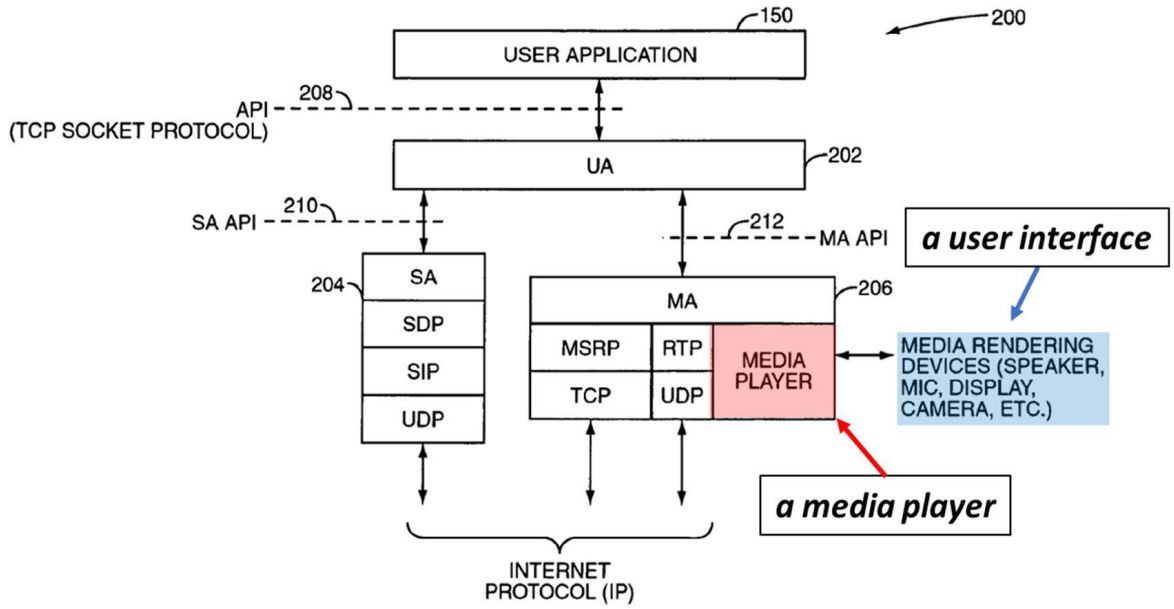


FIG. 3

EX1041, FIG. 3.

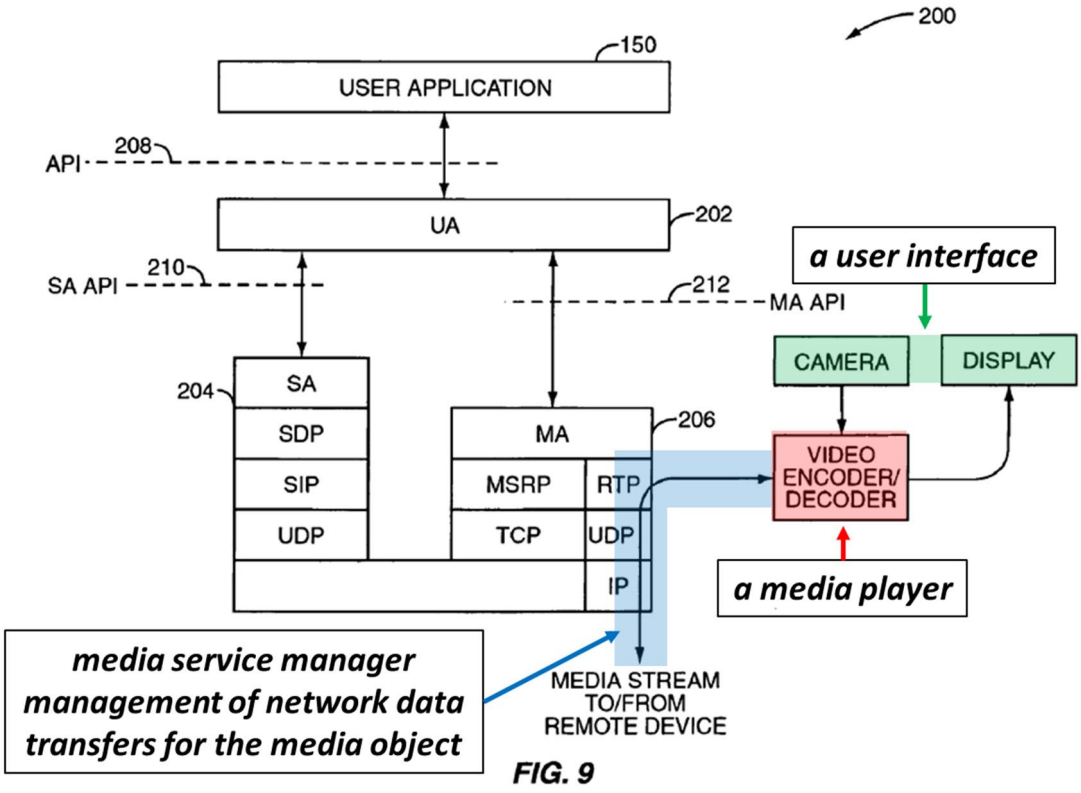


FIG. 9

EX1041, FIG. 9.

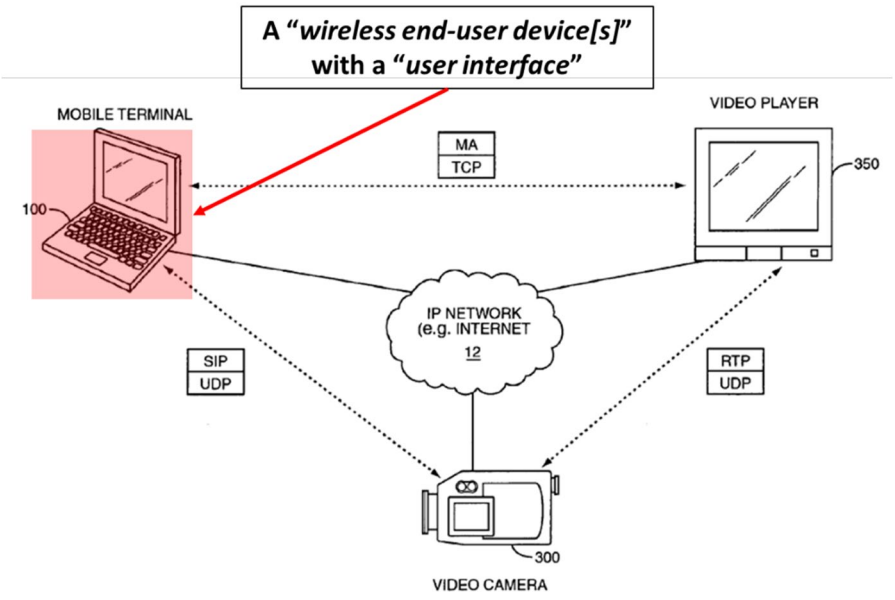


FIG. 11

EX1041, FIG. 11.

9. Claim 13: ***"The wireless end-user device of claim 1, the one or more service classification and measurement agents to further associate one or more traffic flows, comprising the media object network data transfers, with the device application that makes the data transfer request, the device further comprising an enforcement agent to, based on the association between the one or more traffic flows and the device application, enforce an application-based usage control on network data usage by one or more of the device applications."***

118. The '918 patent does not define the feature "***traffic flow***," but generally describes that applications generate "***traffic flows***" (e.g., data transmitted and received when performing activities over a network). EX1001, 62:16-20, 81:36-45, 88:47-53, 89:23-25, 93:42-46, 109:55-59, 120:7-10, FIG. 36.

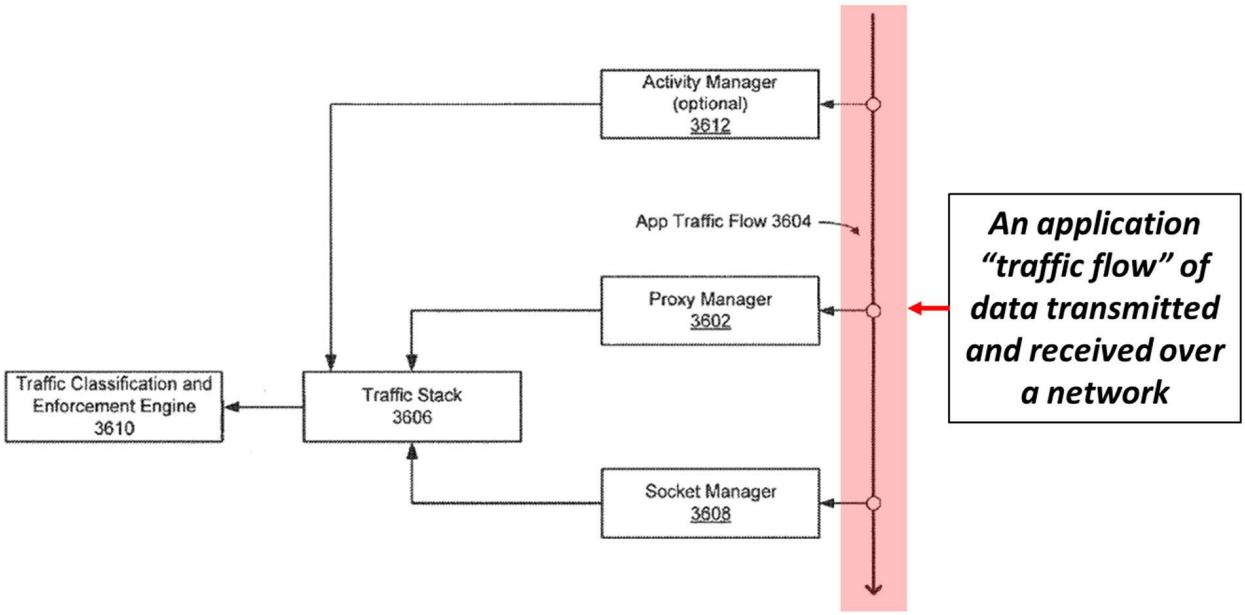


FIG. 36

EX1001, FIG. 36.

119. As described above, *Vadde*'s "policy" based system enforces restrictions on application data usage based on "usage limits" ("***enforce an application-based usage control on network data usage***"). EX1042, ¶¶15-16, 25-26; *see supra* §X.A.4.g (1[f]). In particular, *Vadde*'s restrictor component 122 ("***service classification and measurement agent***" and "***enforcement agent***") monitors transmitted data to determine "attributes 112," which include an "application name" which is given a "Program ID." EX1042, ¶¶24, 32; *see supra* §X.A.4.g (1[f]). By monitoring data traffic by attributes that include application name, restrictor component 122 "***associate[s] one or more traffic flows***" (e.g., data traffic) "***with the device application that makes the data transfer request***" (e.g., the originating application via the application name attribute).

120. Additionally, the data transmitted and received by the *Bennett-Vadde* device's applications ("***traffic flows***") includes streamed media data ("***one or more traffic flows, comprising the media object network data transfers***"). EX1041, ¶¶17-18, 24, 70, 76; *see supra* §X.A.4.e (1[d]).

121. *Vadde*'s restrictor component "monitors the data transmitted and/or received by the applications 110" and applies a "policy" that "prevents the applications 110 from transmitting and/or receiving data in excess of the data usage limits 116" ("***enforce an application-based usage control on network data usage by one or more of the device applications***"). EX1042, ¶22; *see supra* §X.A.4.g

(1[f]). Additionally, because *Vadde*'s enforcement of policies is per application, the application of a policy in response to data usage is "***based on the association between the one or more traffic flows and the device application***." EX1042, ¶22.

10. Claims 14 and 19:

122. In my opinion, claims 14 and 19 are disclosed and/or obvious for the reasons as discussed in the analysis for the corresponding claim listed in the table below.

Claim	Corresponding Claim
14[pre]: A method of operating a wireless end-user device when connected via a wireless modem to a wireless network, the method comprising:	**1[pre]:** A wireless end-user device, comprising **1[a]:** a wireless modem configurable to connect to a wireless network;
14[a]: operating a first network stack Application Programming Interface (API), containing at least one first call accessible to each of a plurality of device applications, the first network stack API callable by each of the plurality of device applications to open and use data flows via a	**1[b]:** a network stack configurable to receive and transmit data via the wireless modem and the wireless network; **1[c]:** a first network stack Application Programming Interface (API), containing at least one first call accessible to each of a plurality of device applications, the first network

network stack coupled to the wireless modem;	stack API callable by each of the plurality of device applications to open and use data packet flows via the network stack, the wireless modem, and the at least one wireless network;
14[b]: operating a second API containing at least one second call accessible to each of the plurality of device applications, the second API callable by each of the plurality of device applications to make a data transfer request for a media object associated with a network resource identifier supplied by the calling device application;	**1[d]:** a second API containing at least one second call accessible to each of the plurality of device applications, the second API callable by each of the plurality of device applications to make a data transfer request for a media object associated with a network resource identifier supplied by the calling device application;
14[c]: operating a media service manager prompted by the second call, the media service manager managing network data transfers for the media object by interfacing with the network stack to retrieve the media object associated with the network resource identifier via the wireless modem and the wireless network; and	**1[e]:** a media service manager prompted by the second call, to manage network data transfers for the media object by interfacing with the network stack to retrieve the media object associated with the network resource identifier via the wireless modem and the wireless network; and

14[d]: associating wireless network data usage for the media object network data transfers with the device application that requests the data transfer for the media object, associating wireless network data usage for respective data packet flows opened and used via the first network stack API with the device application opening such respective data packet flow, and reconciling wireless network data usage for each of the plurality of device applications to track an aggregate wireless network data usage attributable to each of the plurality of device applications via both the first network stack API and the second API.	**1[f]:** one or more service classification and measurement agents to associate wireless network data usage for the media object network data transfers with the device application that requests the data transfer for the media object, to associate wireless network data usage for respective data packet flows opened and used via the first network stack API with the device application opening such respective data packet flow, and to reconcile wireless network data usage for each of the plurality of device applications to track an aggregate wireless network data usage attributable to each of the plurality of device applications via both the first network stack API and the second API.
19[pre]: A wireless end-user device, comprising:	**1[pre]:** A wireless end-user device, comprising
19[a]: a wireless modem configurable to connect to a wireless network;	**1[a]:** a wireless modem configurable to connect to a wireless network;

19[b]: a network stack configurable to receive and transmit data via the wireless modem and the wireless network;	**1[b]:** a network stack configurable to receive and transmit data via the wireless modem and the wireless network;
19[c]: a first network stack Application Programming Interface (API), containing at least one first call to allow device applications to open and use data flows via the network stack, the wireless modem, and the at least one wireless network;	**1[c]:** a first network stack Application Programming Interface (API), containing at least one first call accessible to each of a plurality of device applications, the first network stack API callable by each of the plurality of device applications to open and use data packet flows via the network stack, the wireless modem, and the at least one wireless network;
19[d]: a second API containing at least one second call to allow device applications to make a data transfer request for a media object associated with a network resource identifier supplied by the device application, wherein the data transfer request comprises a network resource identifier that identifies a source of the data to be transferred, a proxy to the source of the data to be	**1[d]:** a second API containing at least one second call accessible to each of the plurality of device applications, the second API callable by each of the plurality of device applications to make a data transfer request for a media object associated with a network resource identifier supplied by the calling device application; **3:** The wireless end-user device of claim 2, wherein the data transfer

transferred, or the media object to be transferred, in particular, wherein the network resource identifier comprises one or more of an Internet Protocol address, a Uniform Resource Locator, a remote file name/address, a stream name, and an object name;	request comprises a network resource identifier that identifies a source of the data to be transferred, a proxy to the source of the data to be transferred, or the media object to be transferred, in particular, wherein the network resource identifier comprises one or more of an Internet Protocol address, a Uniform Resource Locator, a remote file name/address, a stream name, and an object name.
19[e]: a media service manager prompted by the second call, to manage network data transfers for the media object by interfacing with the network stack to retrieve the media object associated with the network resource identifier via the wireless modem and the wireless network, wherein to manage network data transfers for the media object by interfacing with the at least one network stack comprises to map the data transfer request and network resource identifier to one or more data flow connections communicated	**1[e]:** a media service manager prompted by the second call, to manage network data transfers for the media object by interfacing with the network stack to retrieve the media object associated with the network resource identifier via the wireless modem and the wireless network; and **8:** The wireless end-user device of claim 3, wherein to manage network data transfers for the media object by interfacing with the at least one network stack comprises to map the data transfer request and network resource identifier to one or more data

through the device networking stack; and	flow connections communicated through the device network stack.
19[f]: a service classification agent to associate wireless network data usage for the media object network data transfers with the device application that requests the data transfer for the media object, wherein to associate data usage for the media object network data transfers with the device application that makes the data transfer request for the media object comprises to identify at least one of an application name, an application identifier, or a process identifier for the application that makes the data transfer request.	**1[f]:** one or more service classification and measurement agents to associate wireless network data usage for the media object network data transfers with the device application that requests the data transfer for the media object, to associate wireless network data usage for respective data packet flows opened and used via the first network stack API with the device application opening such respective data packet flow, and to reconcile wireless network data usage for each of the plurality of device applications to track an aggregate wireless network data usage attributable to each of the plurality of device applications via both the first network stack API and the second API. **2:** The wireless end-user device of claim 1, wherein to associate wireless network data usage for the media object network data transfers with the device application that makes the data

	transfer request for the media object comprises to identify at least one of an application name, an application identifier, or a process identifier for the application that makes the data transfer request.

123. Claim 14 is a method that implements the features recited in claim 1, with the addition of "operating" the APIs and media service manager. The claims are otherwise substantively identical as I demonstrate in the chart above. In my opinion, the addition of "operating" the APIs and media service manager does not change my analysis discussed above with claim 1 because "operating" is merely the verb associated with how *Bennett*'s signaling agent (SA) 204 "implements SIP and SDP protocols to handle signaling tasks" in response to requests from applications. *See* §§X.A.4.c-X.A.4.e.

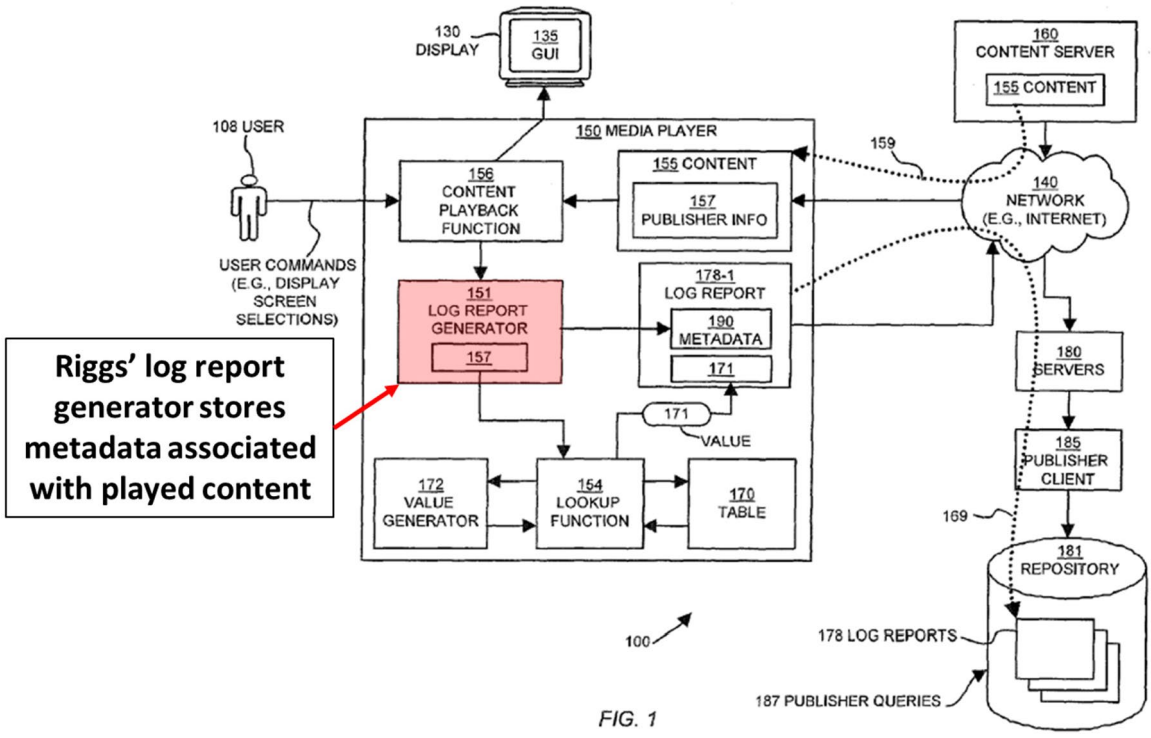
124. Claim 19 is another device claim that is likewise substantively identical to claim 1 and more broadly recites the APIs "allow[ing]" the features recited in claim 1, while incorporating the limitations of dependent claims 3, 8, and 2. The incorporation of the dependent claim language does not change my analysis. *See* §§X.A.4, X.A.5-X.A.7. Furthermore, in my opinion, Claim 19's "service classification agent" is taught and rendered obvious for the reasons explained in

claim 1 with respect to the "one or more service classification and measurement agents." *See* §X.A.4.g.

B. Ground 2: Claims 4-6, 11, and 15-17 Are Obvious over *Bennett* in View of *Vadde* and *Riggs*

1. Overview of *Riggs*

125. *Riggs* discloses a "media player" with an integral "log report generator" that generates a "log report" of "metadata" associated with media. EX1043, 5:20-6:45, FIG. 1. *Riggs*'s metadata includes "application name," "URL," "publisher information," and other information particular to a content source or publisher. EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2.



EX1043, FIG. 1.

2. Motivation to Combine *Bennett-Vadde* and *Riggs* with Reasonable Expectation of Success

126. It would have been obvious to a POSITA to incorporate *Riggs*'s techniques of logging metadata associated with media—to include *Riggs*'s log generators and log reports—into the *Bennett-Vadde* device to log metadata associated with the media played by applications. EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2. *Riggs*'s log reports would have provided a convenient way to track and display per-application data usage determined by *Vadde*'s restrictor component. EX1041, ¶¶2, 25, 76, FIGS. 3, 9, 11; EX1042, ¶21, FIG. 1; EX1043, 6:25-31. A POSITA would have been motivated to make this combination for multiple reasons, including: (1) increased insight into a user's data usage patterns, (2) convenient tracking and sharing of usage data, and (3) increased granularity of data usage tracking on a per-media player basis. EX1042, ¶¶2, 9, 18, 24, 27-28; EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 11:54-13:46, 16:28-33, FIG. 2.

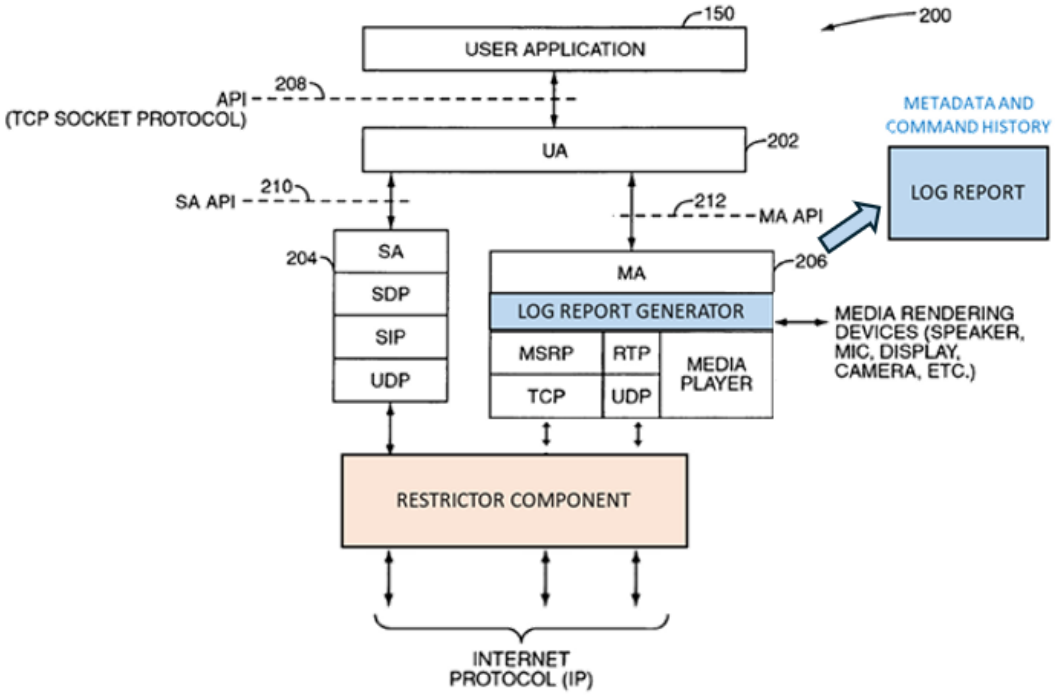
127. Specifically, *Riggs* discloses that, by "record[ing] events such as occurrence of playback commands and related playback information," a publisher can identify corresponding portions of such content that are most appealing to a viewer." EX1043, 1:50-57; 14:27-36. Incorporating *Riggs*'s techniques of logging metadata associated with played content requested via *Bennett's* media agent 206 would have enhanced the ability of third-party publishers to recommend relevant

media to the user of the *Bennett-Vadde* device (e.g., a song from a music streaming service, or a video from a movie streaming service). EX1041, ¶¶24, 76; EX1043, 1:49-57, 14:27-36. Additionally, *Riggs* discloses that, with user consent, this information can be shared between publishers, allowing many such streaming services to gain insight into a user's preferences without the need for a publisher to collect this data organically. EX1043, 1:49-57, 2:35-40, 14:27-36.

128. *Riggs* also teaches that a specific media player may be relevant to streamed media, and to that end, data specifying the type of media player that viewed the media can be collected. EX1043, 11:54-14:26, FIG. 3. This is particularly relevant to *Bennett*, which discloses that "one or more media players" can be included on any given device. EX1041, ¶25. *Riggs*'s techniques allow for the collection of data from multiple media players to be standardized such that this data can be easily correlated and analyzed. EX1043, 11:54-14:26, FIG. 3.

129. In *Bennett-Vadde*, as shown below, it would have been obvious to implement *Riggs*'s log generator as one or more modules associated with *Vadde*'s restrictor component and *Bennett*'s MA206. As *Vadde* recognizes, "program modules … perform particular tasks or implement particular abstract data types" and "may be implemented with any number and organization of such components or modules." EX1042 ¶41. Like *Vadde*'s related report component 124, which provides usage information based on the restrictor component, *Riggs*'s log generator is

complementary to the restrictor component because both track data usage information; it thus would have been obvious to implement *Riggs*'s log generator in the combined system as a module related to *Vadde*'s restrictor component (e.g., as one of the "service classification and measurement agents"). *See* EX1042 ¶¶17, 23. A POSITA would have been motivated to implement *Riggs* this way because the log generator and restrictor component perform and related functions (e.g., tracking application usage and reporting on the same). A POSITA also would have found it obvious to implement *Riggs*'s log generator within *Bennett*'s MA 206 to track media usage, as *Riggs* discloses, because *Bennett* discloses media requests flow via MA 206. *Riggs*'s log generator therefore renders obvious the claimed "one or more service classification and measurement agents" that are "within the media services manager" in the combined system.



EX1041, FIG. 3 (as modified by *Vadde* and *Riggs*).

130. Incorporating *Riggs*'s techniques into the *Bennett-Vadde* device in this manner would have been nothing more than the application of known techniques (e.g., logging metadata according to *Riggs*) to a known structure (e.g., *Bennett-Vadde*'s device) to yield predictable results (e.g., the logging of metadata associated with media played by the *Bennett-Vadde* device's media players within *Bennett*'s MA 206). A POSITA would have expected success in implementing this combination because logging metadata associated with media played by the *Bennett-Vadde* media players using *Riggs*'s techniques simply applies *Riggs*'s teachings—with little modification—to a device that *Riggs* explicitly discloses can implement

its techniques (e.g., a media player). Indeed, the modification would have involved routine programming ability that would have been well within the skill of a POSITA.

3. Claim 4: *"The wireless end-user device of claim 3, wherein to associate wireless network data usage for the media object network data transfers with the device application that makes the data transfer request for the media object further comprises to store an entry comprising the at least one of the application name, the application identifier, or the process identifier for each of the device applications that makes a data transfer request, each stored entry further comprising information about the corresponding network resource identifier for the data transfer request."*

131. *Riggs* discloses a "media player" with a "log report generator" that generates a "log report" of "playback commands" and "metadata" associated with played media ("***store an entry comprising the at least one of the application name, the application identifier, or the process identifier for each of the device applications that makes a data transfer request***"). EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2. Examples of *Riggs* "metadata" include an "[a]pplication [n]ame" ("***application name***") and "media feed URL" ("***process identifier***"). *Id*. *Riggs*'s metadata also includes "a URL associated with the content being played back" ("***each stored entry further comprising information about the corresponding network resource identifier for the data transfer request***"). EX1043, 6:32-45, 10:13-23, 11:18-31, 16:28-33.

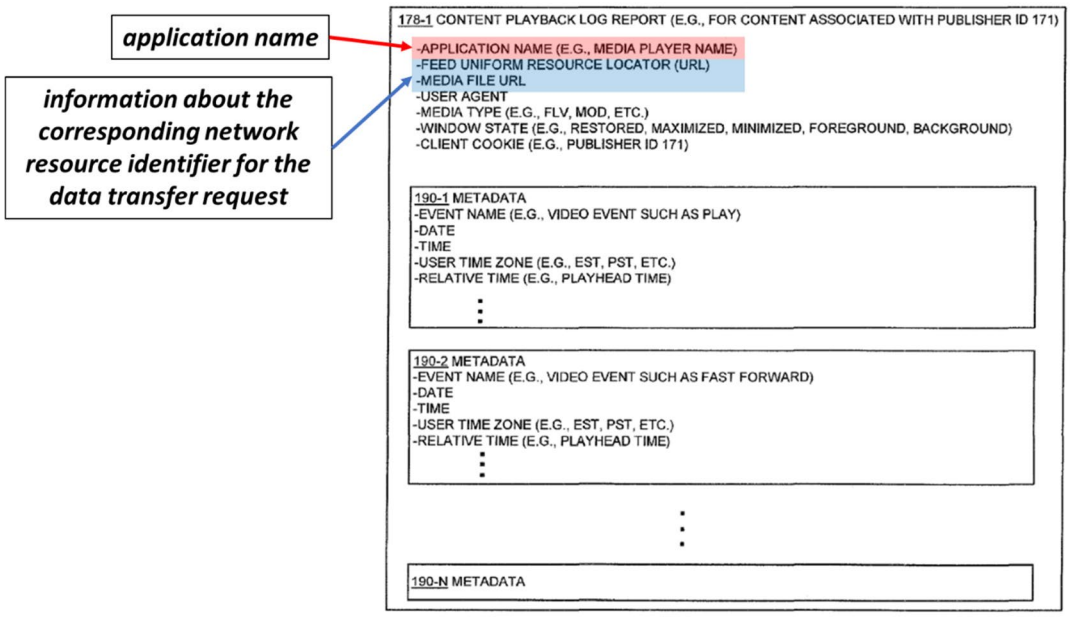
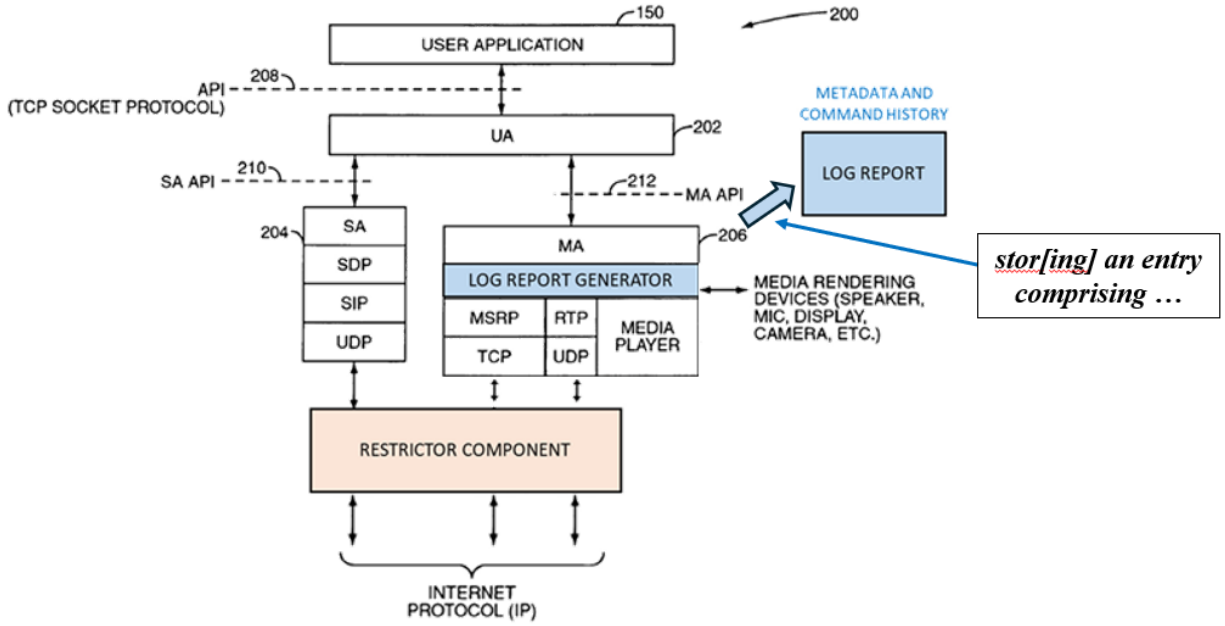


FIG. 2

EX1043, FIG. 2.

132. Additionally, in implementing the *Bennett-Vadde-Riggs* combination, a POSITA would have recognized and found obvious that other information, as described in *Bennett* and *Vadde*, would have been included in the metadata stored in a log report, for example, the Program ID disclosed in *Vadde* (an "***application name***" and "***identifier***"). *See supra* §X.A.5 (2). This would have allowed a POSITA to leverage the existing features of *Bennett* and *Vadde* and adapt them to perform *Riggs*'s techniques in a predictable way that achieves the benefits of *Riggs*'s logging in the context of the media transfers of *Bennett-Vadde*. As described above, in the combination, when monitoring data usage for each of the applications, the *Bennett-Vadde-Riggs* device would have additionally logged "playback commands" and "metadata" describing the retrieved media objects in a "log report," as described in

Riggs ("***to associate wireless network data usage for the media object network data transfers with the device application that makes the data transfer request for the media object further comprises to store an entry comprising …***"). *See supra* §X.B.2. Doing so would have provided the benefits described above, including gaining increased insight into a user's data usage patterns. *See supra* §X.B.2.

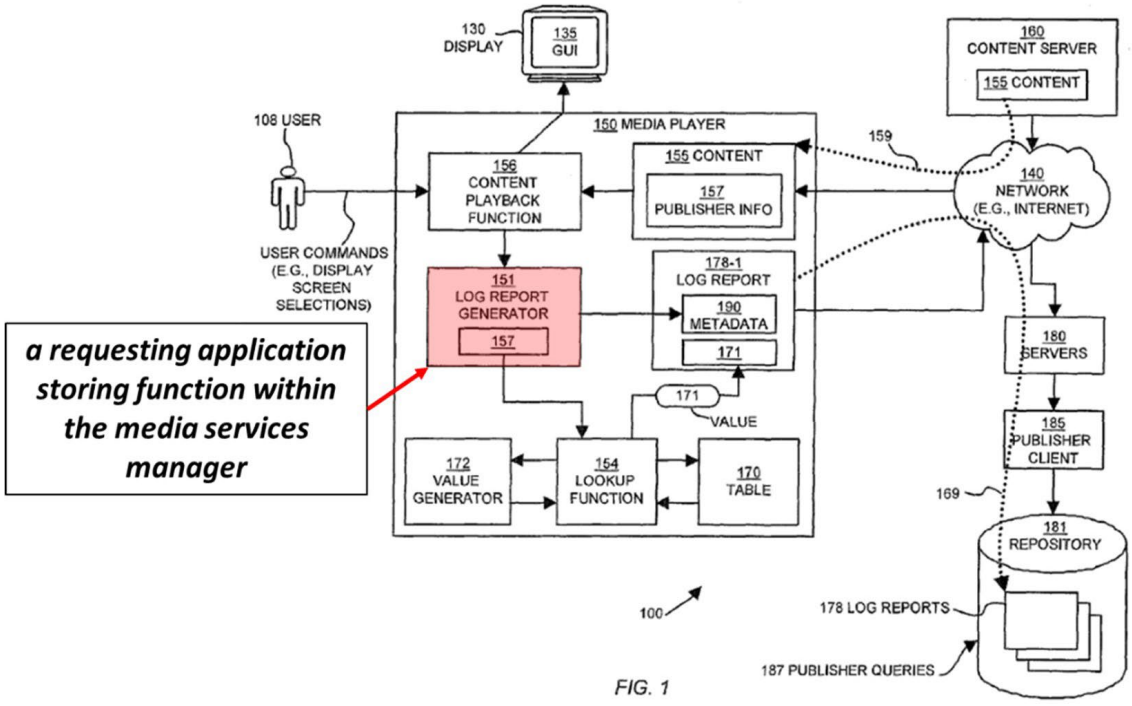


EX1041, FIG. 3 (as modified by *Vadde* and *Riggs*).

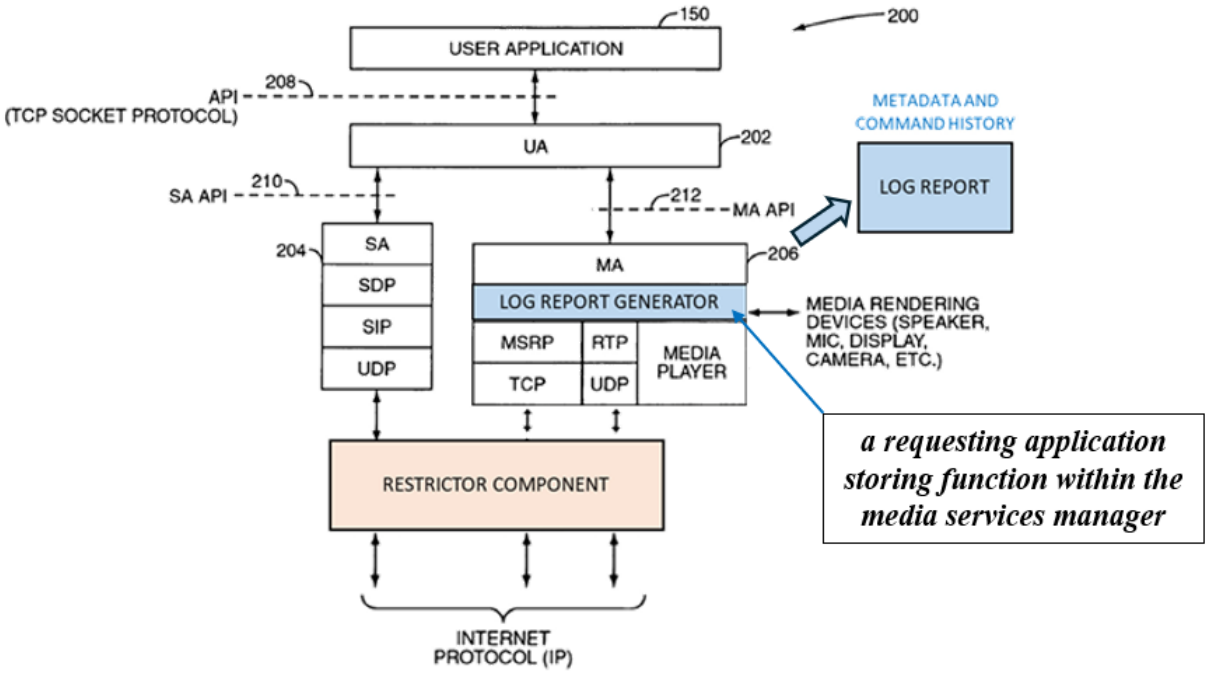
4. Claim 5: *"The wireless end-user device of claim 4, wherein the one or more service classification and measurement agents includes a requesting application storing function within the media services manager."*

133. As described above, *Riggs* discloses a "log report generator" ("***a requesting application storing function***") within the "media player" ("***within the media services manager***") that generates a log report of metadata associated with

played content. EX1043, 5:23-32, 6:32-45, 11:45-49, FIG. 1; EX1001, 112:53-61, 114:27-34 (describing the "requesting-application storing function"); *see supra* §X.B.3 (4) (describing "log reports").



EX1043, FIG. 1.



EX1041, FIG. 3 (as modified by *Vadde* and *Riggs*).

5. Claim 6: *"The wireless end-user device of claim 5, wherein the media service manager is a first media service manager and the requesting application storing function is a first requesting application storing function, the device further comprising a second media service manager of a different type than the first media service manager, the service classification agent including a second requesting application storing function within the second media service manager."*

134. *Riggs* discloses that its techniques can be implemented on multiple media players "350-1" and "350-2" ("***a first media service manager [and] … a second media service manager of a different type than the first media service manager***") that each include "log reports" from a respective "log report generator" ("***a first requesting application storing function [and] … a second requesting application storing function within the second media service manager***"). EX1043,

11:54-13:46. Additionally, Riggs demonstrates that these two media players are "***of a different type***," at least because each of media players 350-1 and 350-2 generate log reports using different "unique values." EX1043, 11:54-13:46.

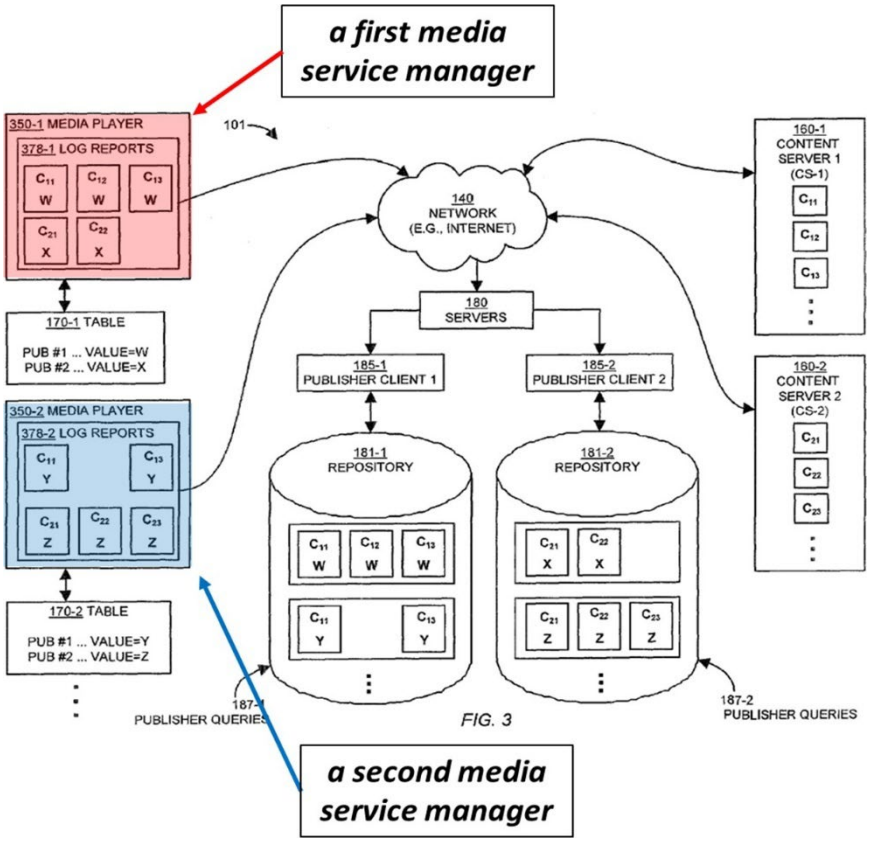
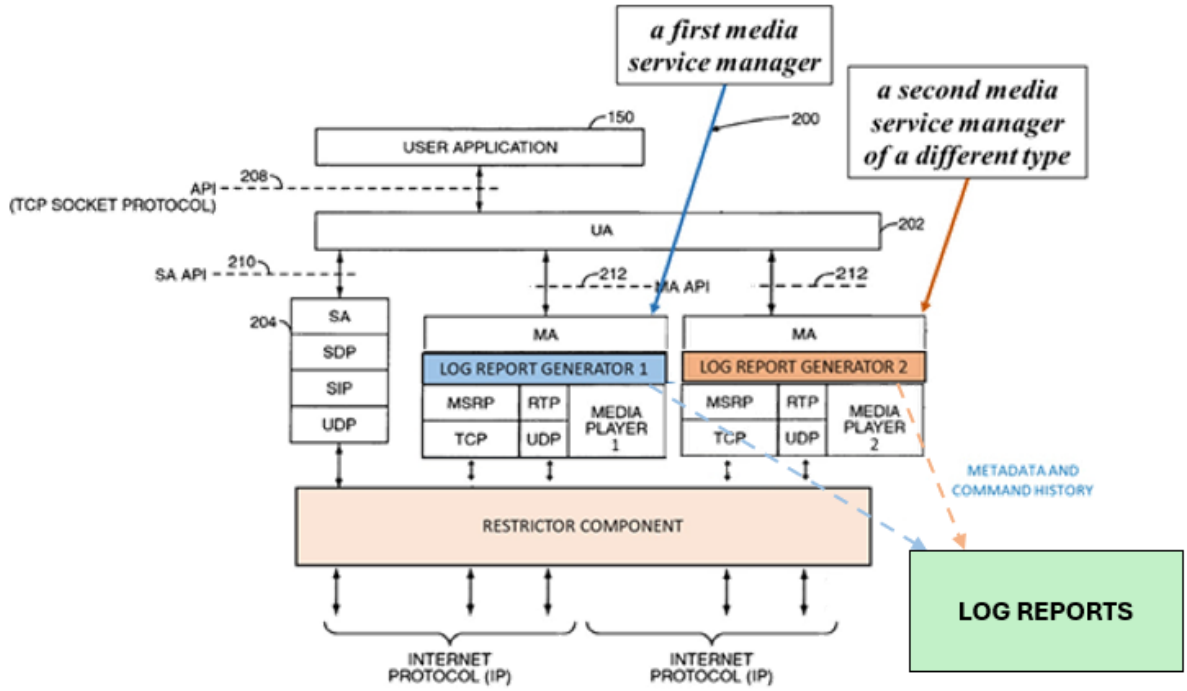


FIG. 3

EX1043, FIG. 3.



EX1041, FIG. 3 (as modified by *Vadde* and *Riggs*).

135. Although *Riggs* does not explicitly disclose that media players 350-1 and 350-2 are operating on the same device, *Riggs* does not otherwise exclude or disparage this arrangement. *See generally*, EX1043. Additionally, *Bennett* describes that its media client includes "one or more media players," and in my opinion, a POSITA would have understood and found obvious that the *Bennett-Vadde-Riggs* device would have included multiple media players, at least because *Riggs* describes that media players can be embodied in "browsers" (a plurality of which can operate on the same device simultaneously). EX1041, ¶25; EX1043, 1:13-24, 1:58-62, 5:33-38. Further, the use of two or more media agents in the *Bennett* device would have enabled simultaneous transmission and reception of media over different protocols,

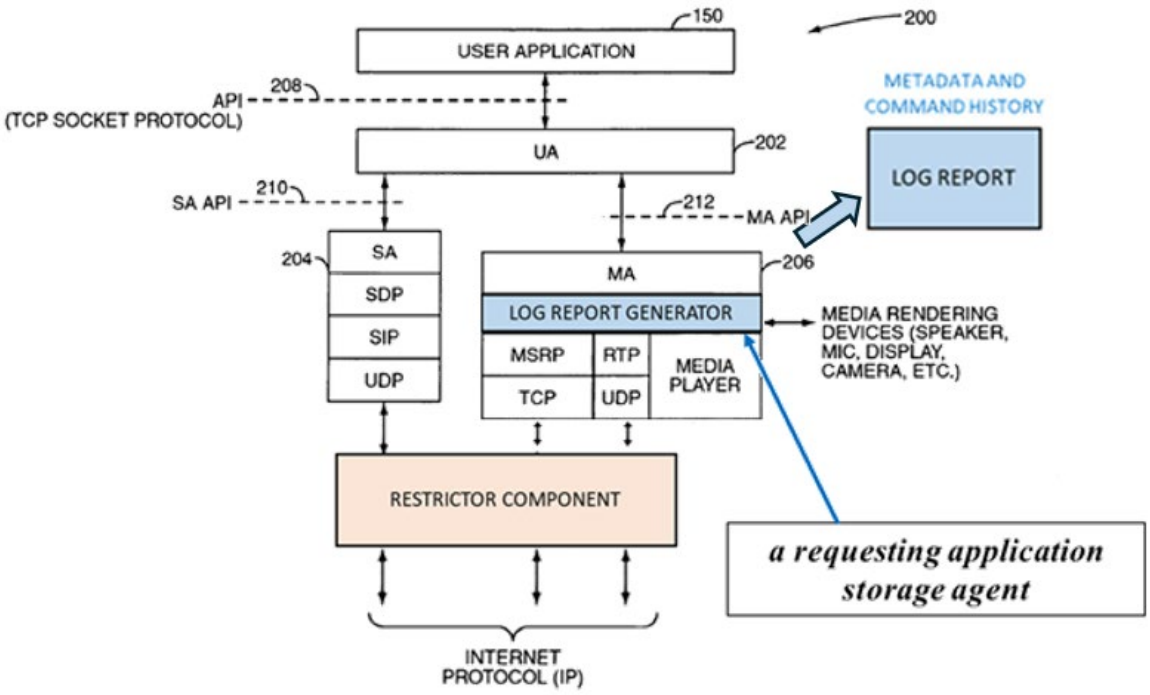
for example, "MSRP" and "RTP" (e.g., MSRP text communications during an active RTP stream). EX1041, ¶¶53-54, 66, 72.

6. Claim 11:

a. 11[a]: *"The wireless end-user device of claim 1, wherein the one or more service classification and measurement agents comprise: a requesting application storage agent to, for each device application that makes a data transfer request using the second API, store application identification information and network resource identification information;"*

136. For the same reasons that *Riggs*'s "log report generator" discloses a "requesting application storing function," it also discloses a "***requesting application storing agent***." *See supra* §§X.B.3 (4) & X.B.4 (5); EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2. In the combination, metadata and data usage data is stored for "each device application that makes a data transfer request using the second API," at least because Vadde's techniques of data monitoring are performed for each application. EX1042, ¶¶22, 24-26, 29-32; *see also* §X.A.4.g (1[f]). A POSITA would have understood that a "function" and "agent" are synonymous terms for software components that perform operations in computer code. Because both of these terms recite software operations that perform "requesting application storing," a POSITA would have understood *Riggs*'s "log report generator" discloses—in unrelated claims—both of these features.

137. Riggs's "log report generator" ("a requesting application storing agent") generates a "log report" of metadata associated with played content ("store application identification information and network resource identification information"). EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2. In the combination, metadata and data usage data is stored for "each device application that makes a data transfer request using the second API," at least because Vadde's techniques of data monitoring are performed for each application. EX1042, ¶¶22, 24-26, 29-32; see supra §X.A.4.g (1[f]).

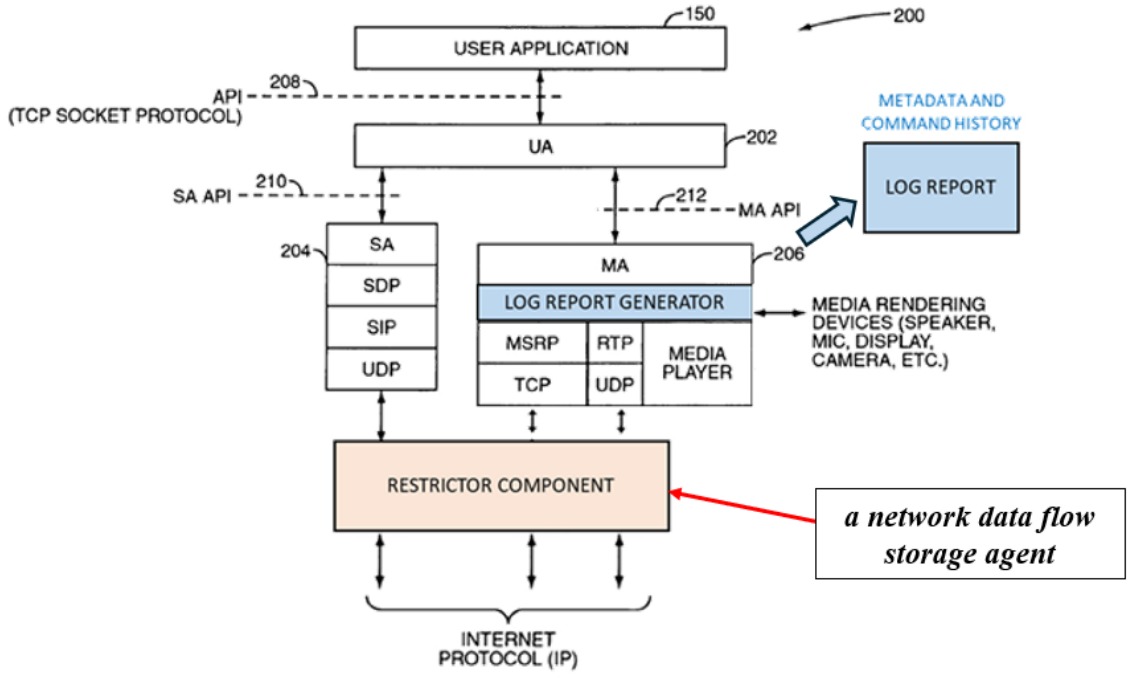


EX1041, FIG. 3 (as modified by *Vadde* and *Riggs*).

b. 11[b]: *"a network data flow storage agent to, for each network data flow associated with the media service manager, identify network data flow identification information; and"*

138. The '918 patent describes that "network data flow identifiers" include "a data flow tag, an IP address, a TCP-IP identifier, a layer 7 identifier, a socket tuple, etc." EX1001, 113:3-9.

139. As described above, *Vadde*'s "restrictor component" ("***a network data flow storage agent***") "monitors the data transmitted and/or received by the applications 110" for "attributes 112" ("***identify network data flow identification information***") that include, among other things, a "network destination" and "port" ("***network data flow identification information***"). EX1042, ¶¶22, 24. Additionally, *Bennett*'s requests include "a network address of a remote host from which media connections will be accepted," for example, a "fully qualified network address" (e.g., an IP address—"***network data flow identification information***"). EX1041, ¶¶34, 50-56, Table-3; *see supra* §X.A.4.e (1[d]). Additionally, as described above, because the restrictor component monitors data transmitted for each application, it monitors "***each network data flow associated with the media service manager***." EX1042, ¶16; *see supra* §X.A.4.g (1[f]).

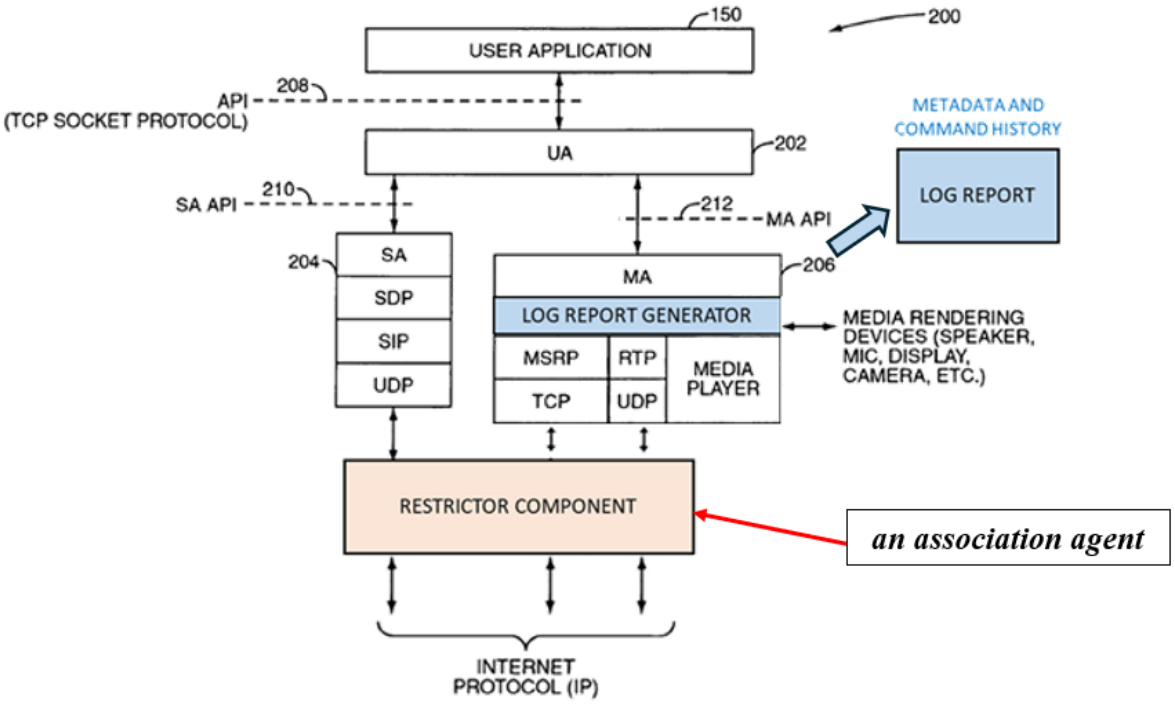


EX1041, FIG. 3 (as modified by *Vadde* and *Riggs*).

c. 11[c]: *"an association agent to match the network data flow identification information for a network data flow with application identification information for the network data transfer associated with the network data flow."*

140. As described above, *Vadde*'s "restrictor component" ("***an association agent***") "monitors the data transmitted and/or received by the applications 110" for "attributes 112." EX1042, ¶¶22, 24; *see supra* §X.A.4.g (1[f]). A POSITA would have recognized and found obvious that *Vadde*'s restrictor component would have "***match[ed] the network data flow identification information for a network data flow***" to "***application identification information associated with [that] network data flow***," at least because *Vadde*'s restrictor component is already capable of

distinguishing per-application usage and this matching would have been needed to produce useful data usage and metadata patterns for each application. EX1042, ¶¶16, 22. Indeed, *Vadde*'s policies are intended to control the usage of a particular application based on that particular application's data usage, and therefore, "***data flow[s]***" and "***application identification information***" would have been obvious to have been "***match[ed]***" to the corresponding application. *Id*.



EX1041, FIG. 3. (as modified by *Vadde* and *Riggs*).

141. The '918 patent discloses the claimed "association" within the same software component it describes for classifying and measuring data usage for each application: via "the service classification and accounting agent." *See* EX1001,

112:37-113:40. *Vadde*'s restrictor component therefore discloses the functionality of the various claimed "***agents***." To the extent claim 11's "***agents***" require separate software components, a POSITA would have found it obvious to implement *Vadde*'s restrictor component as, for example, separate software components in the combined system. *Vadde* discloses that "[e]mbodiments of the invention may be described in the general context of computer-executable instructions," and that "[a]spects of the [its] invention may be implemented with ***any number and organization of such components or modules***." EX1042 ¶41. While *Vadde* depicts its restrictor component 122 as a single component in its figures, *Vadde* discloses that its disclosure is "***not limited*** to the specific computer-executable instructions or the specific components or modules illustrated in the figures and described herein." *Id.* A POSITA therefore would have understood that *Vadde*'s restrictor component may be implemented by "any number" of components ("***agent[s]***") to perform the features of the claimed "***network data flow storage agent***" and "***association agent***." Modifying *Vadde*'s restrictor component 122 as separate components ("***agent[s]***") would have been nothing more than applying *Vadde*'s suggestion to do so, with expected results. The modification would have involved nothing more than routine programming skill with a reasonable expectation of success.

7. Claims 15-17:

142. In my opinion, claims 15-17 are obvious for similar reasons as discussed in the analysis for the corresponding claim listed in the table below.

Claims	Corresponding Claim
15[pre]: A wireless end-user device, comprising:	**1[pre]:** A wireless end-user device, comprising
15[a]: a wireless modem configurable to connect to a wireless network;	**1[a]:** a wireless modem configurable to connect to a wireless network;
15[b]: a network stack configurable to receive and transmit data via the wireless modem and the wireless network;	**1[b]:** a network stack configurable to receive and transmit data via the wireless modem and the wireless network;
15[c]: a first network stack Application Programming Interface (API), containing at least one first call to allow device applications to open and use data flows via the network stack, the wireless modem, and the at least one wireless network;	**1[c]:** a first network stack Application Programming Interface (API), containing at least one first call accessible to each of a plurality of device applications, the first network stack API callable by each of the plurality of device applications to open and use data packet flows via the network stack, the wireless modem, and the at least one wireless network;

15[d]: a second API containing at least one second call to allow device applications to make a data transfer request for a media object associated with a network resource identifier supplied by the device application, wherein the data transfer request comprises a network resource identifier that identifies a source of the data to be transferred, a proxy to the source of the data to be transferred, or the media object to be transferred, in particular, wherein the network resource identifier comprises one or more of an Internet Protocol address, a Uniform Resource Locator, a remote file name/address, a stream name, and an object name;	**1[d]:** a second API containing at least one second call accessible to each of the plurality of device applications, the second API callable by each of the plurality of device applications to make a data transfer request for a media object associated with a network resource identifier supplied by the calling device application; **3:** The wireless end-user device of claim 2, wherein the data transfer request comprises a network resource identifier that identifies a source of the data to be transferred, a proxy to the source of the data to be transferred, or the media object to be transferred, in particular, wherein the network resource identifier comprises one or more of an Internet Protocol address, a Uniform Resource Locator, a remote file name/address, a stream name, and an object name.
15[e]: a media service manager prompted by the second call, to manage network data transfers for	**1[e]:** a media service manager prompted by the second call, to manage network data transfers for the media

the media object by interfacing with the network stack to retrieve the media object associated with the network resource identifier via the wireless modem and the wireless network; and	object by interfacing with the network stack to retrieve the media object associated with the network resource identifier via the wireless modem and the wireless network; and
15[f]: a service classification agent to associate wireless network data usage for the media object network data transfers with the device application that requests the data transfer for the media object, wherein to associate data usage for the media object network data transfers with the device application that makes the data transfer request for the media object comprises to identify and store at least one of an application name, an application identifier, or a process identifier for the application that makes the data transfer request, each stored entry further comprising information about the corresponding network resource identifier for the data transfer request.	**1[f]:** one or more service classification and measurement agents to associate wireless network data usage for the media object network data transfers with the device application that requests the data transfer for the media object, to associate wireless network data usage for respective data packet flows opened and used via the first network stack API with the device application opening such respective data packet flow, and to reconcile wireless network data usage for each of the plurality of device applications to track an aggregate wireless network data usage attributable to each of the plurality of device applications via both the first network stack API and the second API.

	2: The wireless end-user device of claim 1, wherein to associate wireless network data usage for the media object network data transfers with the device application that makes the data transfer request for the media object comprises to identify at least one of an application name, an application identifier, or a process identifier for the application that makes the data transfer request. **4:** The wireless end-user device of claim 3, wherein to associate wireless network data usage for the media object network data transfers with the device application that makes the data transfer request for the media object further comprises to store an entry comprising the at least one of the application name, the application identifier, or the process identifier for each of the device applications that makes a data transfer request, each stored entry further comprising information about the corresponding

	network resource identifier for the data transfer request.
16: The wireless end-user device of claim 15, wherein the service classification agent includes a requesting application storing function within the media services manager.	**5:** The wireless end-user device of claim 4, wherein the one or more service classification and measurement agents includes a requesting application storing function within the media services manager.
17: The wireless end-user device of claim 16, wherein the media service manager is a first media service manager and the requesting application storing function is a first requesting application storing function, the device further comprising a second media service manager of a different type than the first media service manager, the service classification agent including a second requesting application storing function within the second media service manager.	**6:** The wireless end-user device of claim 5, wherein the media service manager is a first media service manager and the requesting application storing function is a first requesting application storing function, the device further comprising a second media service manager of a different type than the first media service manager, the service classification agent including a second requesting application storing function within the second media service manager.

143. Claim 15 is a device claim that is substantively identical to claim 1 and more broadly recites the APIs "allow[ing]" the features recited in claim 1, while

incorporating the limitations of dependent claims 2-4. In my opinion, the incorporation of the dependent claim language does not change my analysis discussed above. *See* §§X.A.4.e, X.A.5-X.A.6, X.B.3. A POSITA would have been motivated with a reasonable expectation of success to incorporate *Riggs*'s log report in *Bennett-Vadde*'s system and method for the same reasons discussed above. *See* §§X.B.2-X.B.3. Moreover, in my opinion, claim 15's "service classification agent" is taught and rendered obvious for the reasons explained in claim 1 with respect to the "one or more service classification and measurement agents" because in the *Bennett-Vadde* combination *Vadde* discloses techniques for "managing data traffic" for each of a plurality of applications using a "policy" based system that enforces restrictions based on "attributes" in the transmitted data and corresponding data "usage limits." *See* §X.A.4.g.

144. Claims 16 and 17 are substantively identical to claims 5 and 6, respectively. Accordingly, in my opinion, the repeated claim language does not change my analysis discussed above. *See* §§X.B.4, X.B.5.

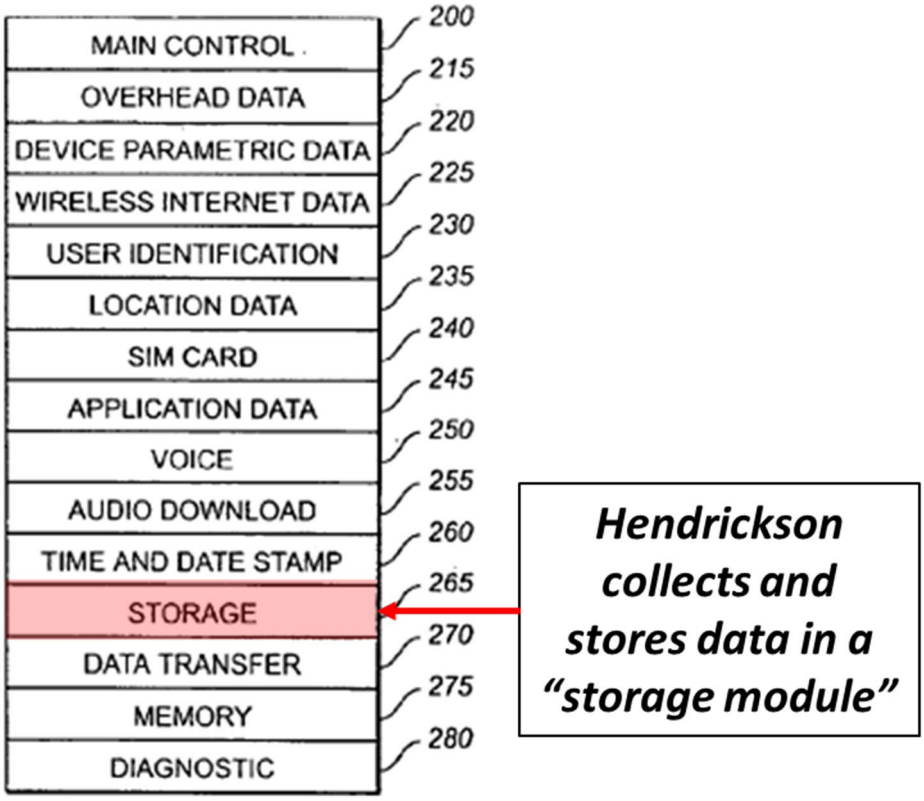
C. Ground 3: Claims 7, 12, and 18 Are Obvious over *Bennett* in View of *Vadde*, *Riggs*, and *Hendrickson*

1. Overview of Hendrickson

145. *Hendrickson* discloses a system for measuring "wireless device and wireless network usage and performance metrics" which collects "device parametric data" and "transmit[s] the collected data via a wireless communication network to

one or more control centers for processing." EX1054, 4:37-5:38, 7:25-8:6, FIGS. 1-2. *Hendrickson* collects and stores its data into a local "storage module 265" prior to transmission to the control center because, as *Hendrickson* notes, there may be "no network connection available to transmit" or "immediate transfer of data [may] result in a poor user experience." EX1054, 12:29-42.

FIG. 2



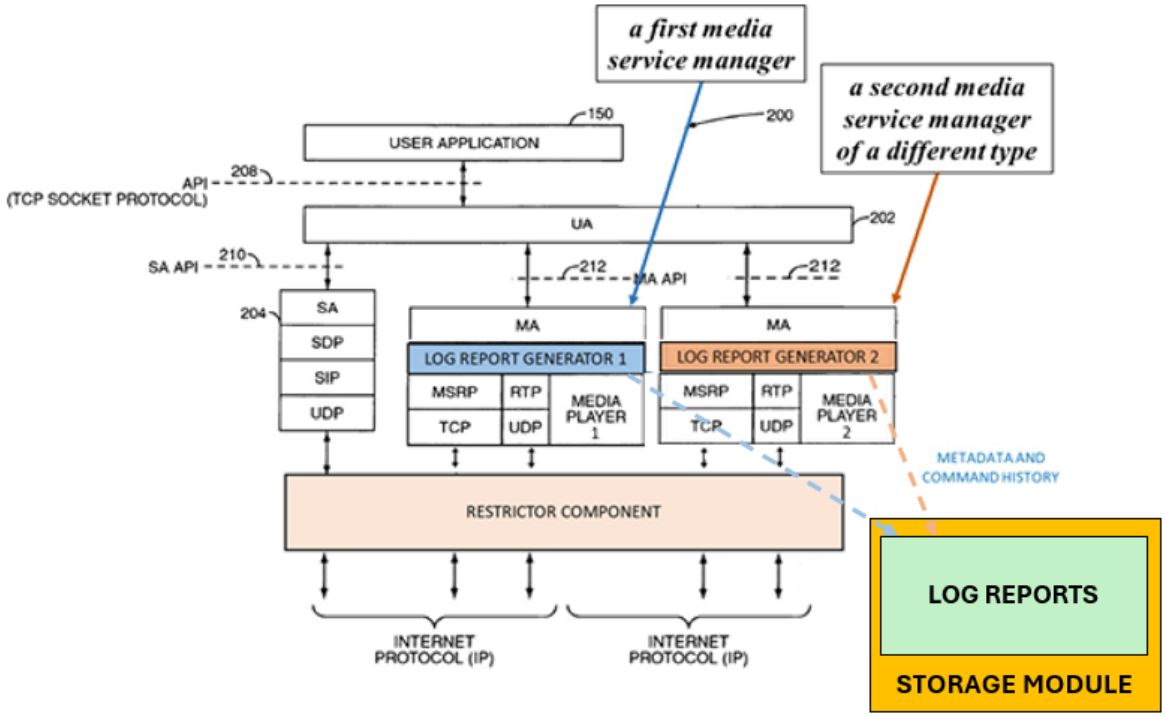
EX1054, FIG. 2.

2. Motivation to Combine *Bennett-Vadde-Riggs* and *Hendrickson* with Reasonable Expectation of Success

146. It would have been obvious for a POSITA to incorporate *Hendrickson*'s techniques of storing data associated with played media content (e.g., using a storage

module) into the *Bennett-Vadde-Riggs* device to store log reports generated by *Bennett-Vadde-Riggs*'s log report generators. EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2; EX1054, 12:29-42; *see supra* §§X.B.3 (4), X.B.5 (6). A POSITA would have recognized and found obvious that generated log reports would have needed to have been stored locally on the device at least temporarily, at least to preserve the data across device operating cycles in the absence of a network connection. EX1054, 12:29-42 (noting that storing data is prudent when "there is no network connection available to transmit" or "immediate transfer of data would result in a poor user experience"); *see supra* §§X.B.3 (4), X.B.5 (6). A POSITA would have found it obvious to implement *Hendrickson*'s storage module 265 in the combined system as organized records that can be accessed by the device (e.g., as a database) in view of *Hendrickson*'s and *Riggs*'s disclosures of sending this data to a remote database for long-term storage. EX1043, 7:42-8:14; EX1054, 12:29-42, 18:53-19:4. A POSITA would have recognized the benefits of doing so, including facilitating the synchronization of data with remote storage by transferring data in the format/structure of the permanent storage database, allowing users to access and search the records storage in the storage module 265 while on the device. This is particularly the case if the data is to be stored in the combined *Bennett-Vadde-Riggs-Hendrickson* for long periods, it would enable a user to access and use the data stored in the storage module.

147. Incorporating *Hendrickson*'s techniques into the *Bennett-Vadde-Riggs* device would have been nothing more than the application of known techniques (e.g., storing data in the form of log reports) to a known structure (e.g., *Bennett-Vadde-Riggs*'s device) to yield predictable results (e.g., the storing of log reports associated with media played by the *Bennett-Vadde-Riggs* device). In my opinion, a POSITA would have expected success in implementing this combination because the storing of *Bennett-Vadde-Riggs*'s log reports using *Hendrickson*'s techniques simply applies *Hendrickson*'s teachings—with little modification—to a device that performs the same functions that *Hendrickson* envisions within its own disclosure (e.g., a system that logs metadata associated with content played by a media player). Indeed, the modification would have (1) involved routine programming ability that would have been well within the skill of a POSITA, and (2) leveraged the existing infrastructure of the base references in a way that was already well known in the art (e.g., using the device memory that *Bennett*, *Vadde*, and *Riggs* all individually disclose and render obvious). EX1041, ¶29 ("[t]he host device includes memory in which to store code implementing the present invention"); EX1042, ¶¶11-14, 17 (describing a device "memory area 108"); EX1043, 14:37-15:2, FIG. 4 (describing a "computer system" with a "memory system 112").



EX1041, FIG. 3 (as modified by *Vadde*, *Riggs*, and *Hendrickson*).

3. Claim 7: *"The wireless end-user device of claim 6, further comprising a usage and classification database, the one or more service classification and measurement agents to receive application association information stored by the first and second requesting application storing functions, and to maintain the usage and classification database based in part on the received application association information."*

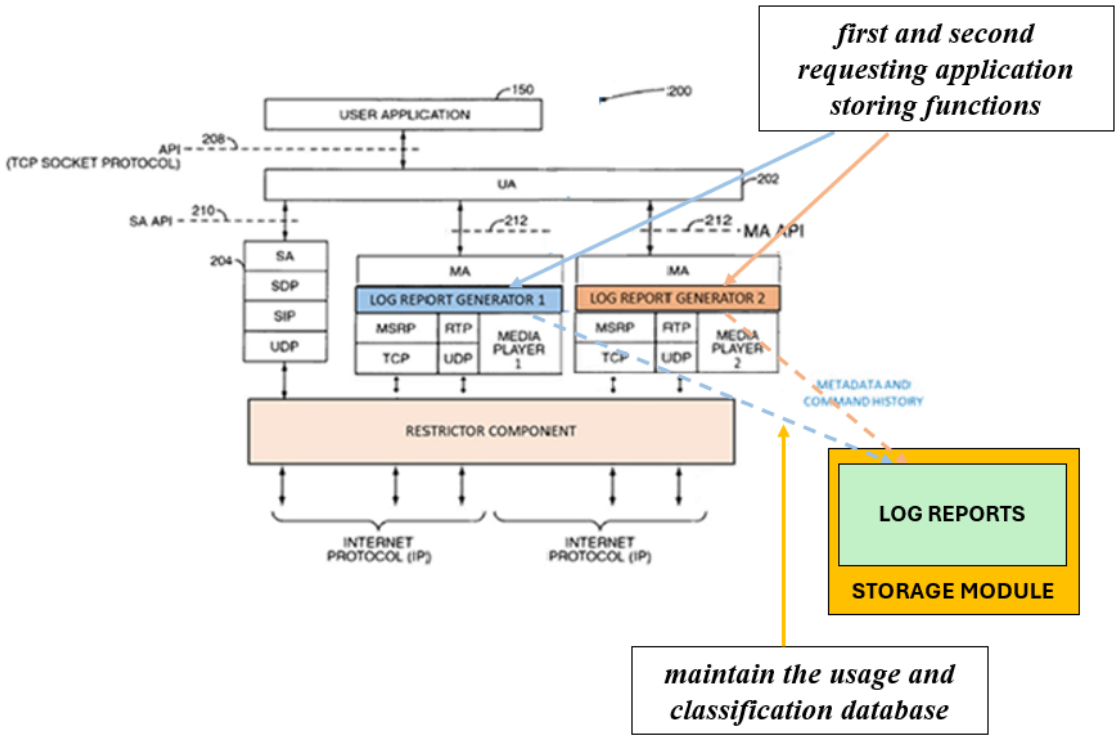
148. The '918 patent does not define a "***usage and classification database***" but rather describes that "the results of the service usage classification or accounting can be stored in a local device (or operating system) database" and that the stored information "can be provided to other applications, operating system service functions, other device software functions, or network-based service classification or accounting functions." EX1001, 113:46-55. As described above, *Riggs*'s "log

report generators" ("***first and second requesting application storing functions***") generate "log reports" of "metadata" associated with content played on a media player ("***application association information***"). EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2; *see supra* §§X.B.3 (4), X.B.5 (6). As described above, *Riggs*'s log report generators are "***one or more service classification and measurement agents***" that "***receive application association information stored by the first and second requesting application storing functions***." *See supra* §X.B.4 (5).

149. In my opinion, a POSITA would have recognized and found obvious that the generated log reports would have been stored as an organized collection of data and/or in a local repository on the device (e.g., a "***database***"), at least because this storage would have been needed to preserve the data across device operating cycles in the absence of a network connection. EX1054, 12:29-42. Moreover, the *Bennett-Vadde-Riggs* device would have included a means for electronic storage of files, like log reports. EX1041, ¶29 ("[t]he host device includes memory in which to store code implementing the present invention"); EX1042, ¶¶11-14, 17 (describing a device "memory area 108"); EX1043, 14:37-15:2, FIG. 4 (describing a "computer system" with a "memory system 112"). *Riggs*'s log reports therefore renders obvious the "***usage and classification database***."

150. To the extent that a POSITA would not have found it obvious that *Riggs*'s log reports are stored locally on the device prior to transmission, *Hendrickson* discloses a "storage module 265" ("***usage and classification database***") on a mobile device as part of a system for measuring "usage and performance metrics" for applications operated on the mobile device. EX1054, 6:40-7:11, 12:29-42, FIG. 2. The storage module 265 is responsible for "collecting information from each data module and encrypting, compressing, and storing the data in log file format in the non-volatile memory locations of the wireless device" and "temporarily stor[ing] data before being handled by the Data Transfer Module" which transmits the collected data (a "***usage and classification database***" that stores data "local[ly]" such that data "can be provided to other applications, operating system service functions, other device software functions, or network-based service classification or accounting functions"). EX1054, 12:29-42; EX1001, 113:46-55. In the combination, the storage module 265 would have stored log reports generated by the *Bennett-Vadde-Riggs* log report generators, prior to these reports being transmitted for external consumption ("***maintain the usage and classification database based in part on the received application association information***"). EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2; EX1054, 12:29-42 (noting that storing data is prudent when "there is no network connection

available to transmit" or "immediate transfer of data would result in a poor user experience"); *see supra* §§X.B.3 (4), X.B.5 (6), VII.C.



EX1041, FIG. 3 (as modified by *Vadde*, *Riggs*, and *Hendrickson*).

4. Claim 12:

a. 12[a]: *"The wireless end-user device of claim 1, further comprising: a local database to store data usage, including data usage for network data transfers managed by the media service manager on behalf of a device application, the stored data usage classified by device application;"*

151. As described above, a POSITA implementing the *Bennett-Vadde-Riggs* combination would have recognized and found obvious that *Riggs*'s log reports would have been a convenient way to log data usage determined from *Vadde*'s

restrictor component. *See supra*, §X.B.2. In the combination with *Hendrickson*, *Riggs*'s log report generators would generate log reports which are then stored in *Hendrickson*'s "storage module 265," which would have been obvious to implement as "***a local database to store data usage***") that stores a record of metadata and data usage associated with played content for each application in the combined *Bennett-Vadde-Riggs* system ("***data usage for network data transfers managed by the media service manager on behalf of a device application***"). EX1043, 1:58- 2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2; *see supra* §§ X.A.4.f (1[e]) (describing *Bennett-Vadde* disclosure of a media service manager managing network data transfers for the media object), X.C.2 (describing implementation as a "database"), X.C.3 (7) (explaining storage in a "database"). Additionally, both *Vadde* and *Riggs* disclose that this information is "***classified by device application***" (particularly, using an application name or Program ID). EX1042, ¶14; EX1043, 11:18-31.

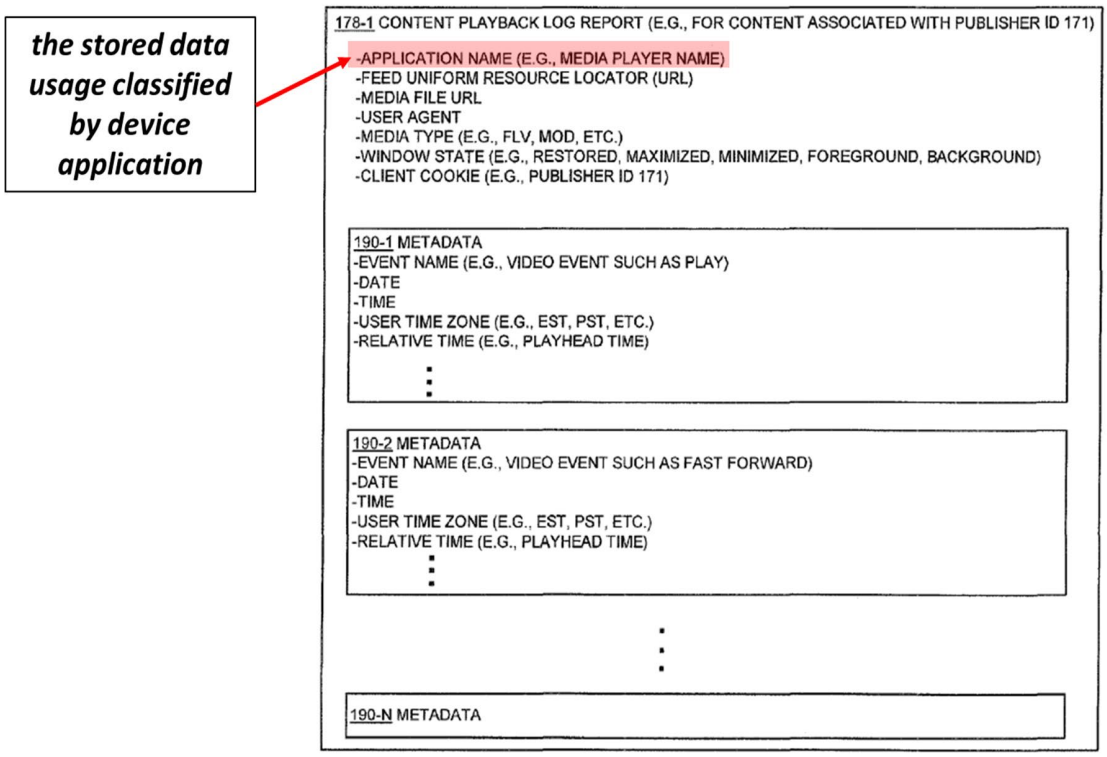


FIG. 2

EX1043, FIG. 2.

b. 12[b]: *"a user interface"*

152. This claim limitation is obvious for the reasons discussed above. *See supra* §X.A.8 (9).

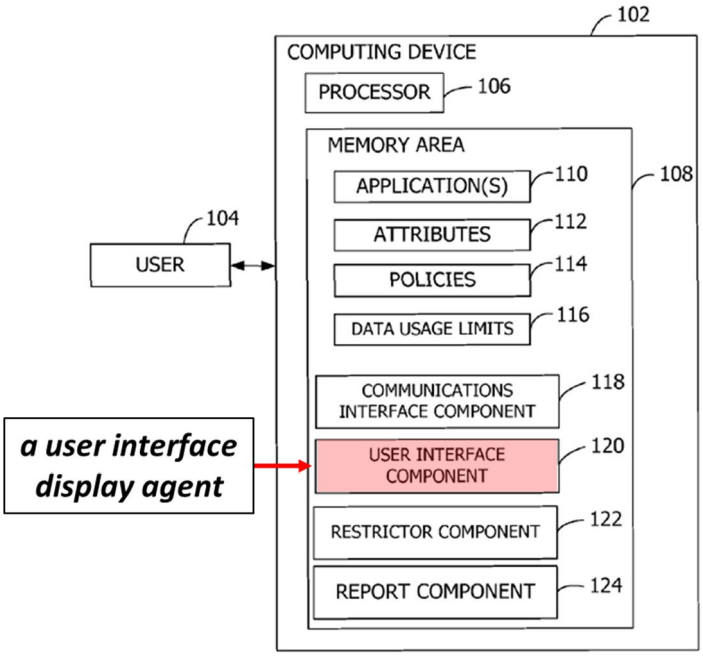
c. 12[c]: *"a user interface display agent to display the data usage classified by application to a user."*

153. *Vadde* discloses a "user interface component 120" ("***a user interface display agent***") that is used for "displaying data to the user" ("***display the data usage classified by application to a user***"). EX1042, ¶21, FIG. 1.

154. In the *Bennett-Vadde-Riggs-Hendrickson* combination, generated log reports for each application—including metadata and data usage—would have been

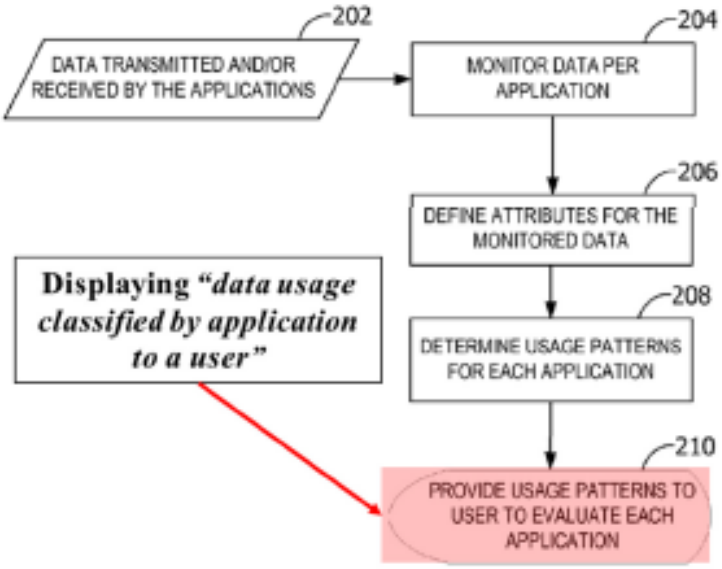
displayed to the user via the "user interface component 120" and a "user interface" ("***the data usage classified by application to a user***"). Displaying log reports to the user would have enabled the user to monitor their per-application data usage and *Vadde* specifically notes that "usage patterns" (including data usage) are provided to the user "to evaluate each application" ("***the data usage classified by application to a user***"). EX1042, FIG. 2. Displaying log reports to the user would have also been a convenient way for the user to verify what information was being shared with publishers (particularly because *Riggs* discloses the need to obtain a user's "consent" before sharing data). EX1043, 1:39-48, 14:27-36.

FIG. 1



EX1042, FIG. 1.

FIG. 2



EX1042, FIG. 2.

5. Claim 18: ***"The wireless end-user device of claim 17, further comprising a usage and classification reconciliation agent and usage and classification database, the usage and classification reconciliation agent to receive application association information stored by the first and second requesting application storing functions, and to maintain the usage and classification database based in part on the received application association information."***

155. As an initial matter, claim 17 repeats the same features recited in claim 7 above, with the substitution of a "usage and classification reconciliation agent" for "the one or more service classification and measurement agents." See supra §X.C.3 (7). The '918 patent does not define a "usage and classification reconciliation agent," but instead describes a "usage/classification reconciliation engine 3216" that "can

track usage of [an] app, classify app, and to the extent there is disagreement at different system locations, reconcile usage in accordance with rules." EX1001, 117:12-24.

156. As described above, *Riggs*'s "log report generators" ("***first and second requesting application storing functions***") generate "log reports" which are then stored in *Hendrickson*'s "storage module 265" ("***a usage and classification database***") which stores a record of "metadata" associated with played content ("***maintain the usage and classification database based in part on the received application association information***"). EX1043, 1:58-2:5, 6:32-45, 10:13-23, 11:18-31, 16:28-33, FIG. 2; *see supra* §X.C.3 (7).

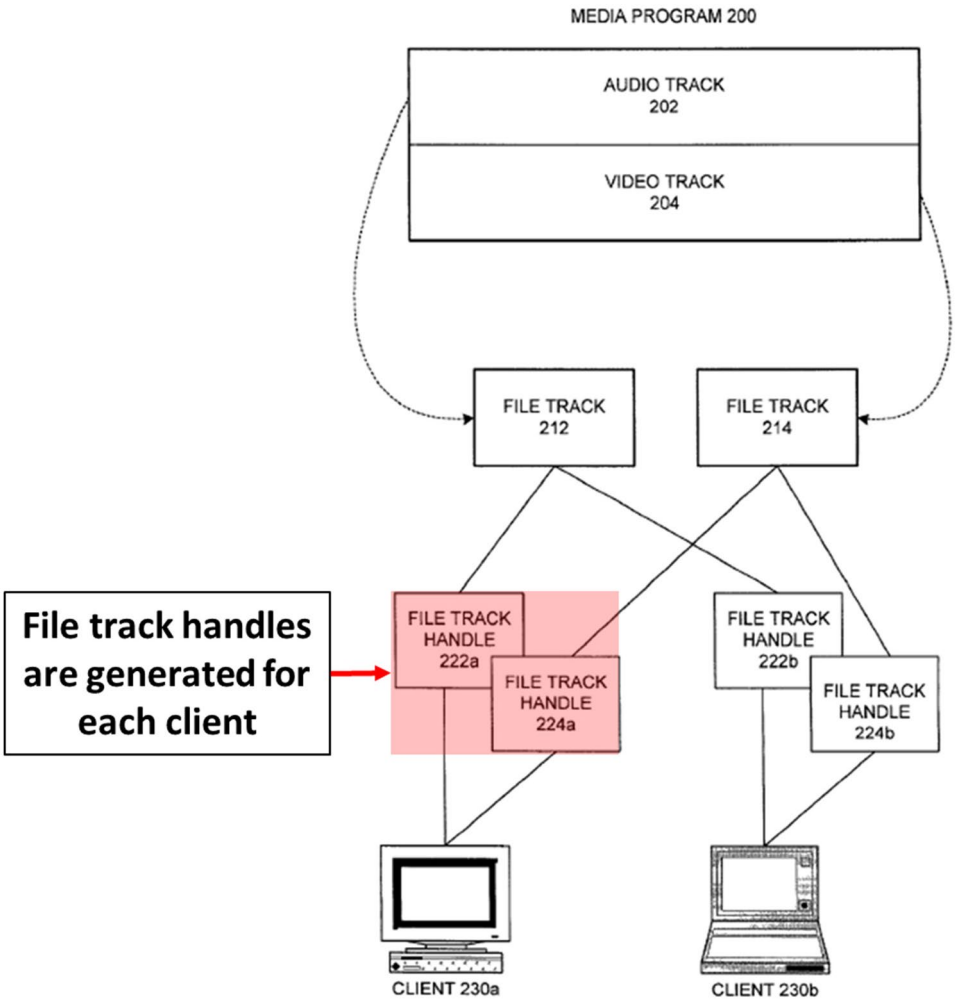
157. *Vadde*'s restrictor component 122 "monitors the data transmitted and/or received by the applications 110 and determines whether the data usage limits 116 corresponding to each of the applications 110 have been exceeded or are about to be exceeded." EX1042, ¶¶22, 24-26, 29-32. As described above, in the combination, log reports are generated based on metadata determined from *Riggs*'s log report generators and usage data determined by *Vadde*'s restrictor component, both of which determine this information for each application. *See supra* §§X.A.4.g (1[f]), X.A.5 (2), X.B.3 (4). Accordingly, *Riggs*'s log report generators, in combination with *Vadde*'s restrictor component, are also "***a usage and classification reconciliation agent***" that "***receive[s] application association information stored***

by the first and second requesting application storing functions" and "***maintain[s] the usage and classification database based in part on the received application association information***" at least because *Riggs*'s log report generators receive data regarding application data usage for each application from *Vadde*'s restrictor component. *See supra*, §§X.A.4.g (1[f]), X.A.5 (2), X.B.3 (4); EX1001, 117:12-24.

D. Ground 4: Claim 10 Is Obvious over *Bennett* in View of *Vadde*, *Riggs*, and *Srikantan*

1. Overview of *Srikantan*

158. *Srikantan* discloses techniques for "streaming a media track to multiple clients using a single copy of the track's metadata, rather than making separate copies of the metadata for each stream." EX1055, Abstract, ¶¶40-47. To manage the streaming of the track to multiple clients, *Srikantan* generates a "file track handle" that "acts as an interface between its client stream … and the single instance of media metadata." EX1055, ¶¶8-9, 44-46, 51-54, 59, 71, FIG. 2. *Srikantan* also discloses that the receiving client "may issue commands to control the stream—e.g., to rewind or fast forward to locate a particular part of the media, to pause the streaming, etc." EX1055, ¶¶24, 29, 60, 73.



EX1055, FIG. 2.

2. Motivation to Combine *Bennett-Vadde-Riggs* and *Srikantan* with Reasonable Expectation of Success

159. It would have been obvious for a POSITA to incorporate *Srikantan*'s techniques, to include a media streaming server that generates file track handles for media files streamed to the *Bennett-Vadde-Riggs* device. EX1041, ¶¶25, 76, FIG. 3; EX1055, ¶¶8-9, 44-46, 51-54, 59, 71, FIG. 2. A POSITA would have recognized and found obvious that generating multiple copies of metadata associated with a

media file is resource intensive, and *Srikantan*'s techniques would have reduced server-side computing requirements by reducing the need to generate multiple copies of metadata for a streamed media file. EX1055, ¶¶4, 7-9, 19-20, 24-26. Also, as *Srikantan* notes, sharing a single file handle among multiple clients "can lead to a great deal of contention among the client streams as each one attempts to seek to (i.e., find) and extract a different media segment or sample"—streams that a POSITA knows cost precious bandwidth. EX1055, ¶¶5-6; EX1042, ¶34; EX1046, 7:40-50, 14:61-67, Table-2 (describing "real-time audio and video" applications as "[h]igh bandwidth" applications); *see supra* §X.A.3. To that end, a POSITA would have recognized the bandwidth savings that would be gained from eliminating conflicting streaming sessions using *Srikantan*'s techniques. *Id*.

160. Incorporating *Srikantan*'s media streaming server and file track handle generation techniques into the *Bennett-Vadde-Riggs* device would have been nothing more than the application of known techniques (e.g., generating a file track handle for a media file) to a known structure (e.g., media streamed by the *Bennett-Vadde-Riggs* device) to yield predictable results (e.g., the *Bennett-Vadde-Riggs* device referencing the *Srikantan* file track handle when streaming a media file). A POSITA would have expected success in implementing this combination because generating and using file handles to reference media streamed on the *Bennett-Vadde-Riggs* device using *Srikantan*'s techniques simply applies *Srikantan*'s teachings—

with little modification—to a system that *Srikantan* itself envisions within its own disclosure (e.g., a device that streams media accessed from a server). EX1055, ¶¶23-33. Indeed, the modification would have (1) involved routine programming ability that would have been well within the skill of a POSITA, (2) introduced concepts that were already well known in the industry ("file handles," or "file descriptors"), and (3) required little to no modification to *Srikantan*'s envisioned system. EX1055, Cover, ¶¶16-18, 83. In my opinion file handles were, and still are, a well-known technique to ensure continuity of a file being shared in a server-client system, as evidenced by *Srikantan* pre-dating the Critical Date by almost a decade. EX1055, Cover; EX1058, 3 (defining a file handle as "[a] temporary designation an operating system assigns to an opened file during any given session"); EX1059, 3 (defining a file descriptor as "[a] value used to identify an open file for the purpose of file access").

3. Claim 10:

a.* 10[a]: *"The wireless end-user device of claim 9, the media service manager to receive, from the application launching the data transfer request, a network resource indicator that identifies the media object"

161. As described above, *Bennett*'s "CALL requests," sent by the application ("***from the application launching the data transfer request***") to the UA 202 include "a network address of a remote host from which media connections will be accepted" ("***a network resource identifier***"), and this information is forwarded to

the MA 206 via an "OPEN request" ("***the media service manager to receive***"). EX1041, ¶¶34, 50-56, Table-3; *see supra* §X.A.4.e (1[d]). As discussed above, *Riggs* discloses that "the global address of content" ("***a network resource indicator that identifies the media object***") is "typically" provided "in the form of a Uniform Resource Locator (URL)" (e.g., a "feed URL"—"***the media object***"—provided by the requesting application). EX1043, 1:25-35, 5:54-62, 6:38-43; *see supra* §X.A.6 (3).

b. 10[b]: *"return to the application a media object handle descriptor"*

162. *Srikantan* discloses that a "file track handle" ("***a media object handle descriptor***") is created when a client requests a media file (a "stream"). EX1055, ¶¶8-9, 44-46, 51-54, 59,71, FIG. 2; EX1058, 3; EX1059, 3. In the *Bennett-Vadde-Riggs-Srikantan* combination, a POSITA would have recognized and found obvious that in response to the "OPEN request" of the UA 202, the MA 206 ("***the media service manager***") would have retrieved *Srikantan*'s "file track handle" ("***a media object handle descriptor***") for the stream referenced in the network address of the OPEN request. EX1041, ¶¶34-35, 50-56, 86, Table-3. Once retrieved, the MA 206 would have returned the file track handle ("***a media object handle descriptor***") to the UA 202 via the "OPEN response," which would then return the file track handle to the application via the "CALL response" ("***return to the application a media object handle descriptor***"). *Id*.

163. In my opinion, a POSITA would have leveraged the MA 206 to return the file handle of *Srikantan* to the application because this arrangement is consistent with the responsibilities of the MA 206, as disclosed in *Bennett*. For example, *Bennett* describes that the MA 206 "manages media connections" to include "Real-Time Transport Protocol (RTP)" sessions. EX1041, ¶¶25, 34, 54-55. Specifically, the OPEN request sent to the MA 206 is an instruction "to initiate an RTP session," and in response, the MA 206 retrieves information from the host ("the network address of the host and port opened for the RTP connection"). Similarly, *Srikantan* describes that its media servers use "RTP (Real-Time Transport Protocol) to deliver the stream to the client." EX1055, ¶¶27, 46, 72. For at least this reason a POSITA would have recognized that when establishing an RTP session with a host server, the MA 206 ("***the media service manager***") would have retrieved a file handle ("***media object handle descriptor***") for the requested stream, along with the "network address of the host and port," to share with the application streaming the file in the RTP session, such that the application could manipulate the stream in the session while informing the hosting server of the applied commands (e.g., "pause," "rewind," or "fast-forward"). EX1041, ¶¶77, 80, 86, Table-3; EX1055, ¶¶24, 29, 60, 73.

164. Further, this file handle would have been specifically "***return[ed] to the application***" because, as *Bennett* notes, "the user application 150 may want to receive the media stream" and, further, "the user application 150 can direct how

media or messages are routed," to include specifying commands like "PAUSE" or "RESUME." EX1041, ¶¶76-77.

c. 10[c]: *"call a proxy service to perform one or more network data transfers comprising the media object"*

165. *Srikantan* discloses a "media streaming server" ("***a proxy service***") that "is configured to stream QuickTime media and/or other forms of media, in a unicast or multicast mode, over a proprietary or publicly accessible network such as the Internet" ("***one or more network data transfers comprising the media object***"). EX1055, ¶¶27-33, FIG. 1. Additionally, because *Bennett*'s MA 206 "manages media connections" to include "Real-Time Transport Protocol (RTP)" sessions—the same RTP sessions *Srikantan* discloses its media streaming server maintains for streaming media—the MA 206 would "***call***" the media streaming server hosting the stream ("***a proxy service***") to "***perform one or more network data transfers comprising the media object***" using an RTP session. EX1041, ¶¶25, 34, 54-55; EX1055, ¶¶27, 46, 72; *see supra* §X.D.3.b (10[b]).

166. Additionally, in my opinion, a POSITA would have recognized and found obvious that *Srikantan*'s "media streaming server" is "***a proxy service***" (such that the *Bennett-Vadde-Riggs-Srikantan* device "***calls a proxy service to perform one or more network data transfers comprising the media object***"), at least because *Srikantan* discloses that the media streaming server can "redirect to clients media that it receives from another entity, such as a live event, a video camera, a broadcast

from another server (e.g., server 130)." EX1055, ¶¶25, 32, 39, FIG. 1. *Srikantan* discloses that in this mode, the media streaming server "acts as a client" when it receives content from another server to send to the requesting device is a mode of operation consistent with a "***proxy***" as of the Critical Date. EX1056, 3-4 (describing a "proxy" as "an intermediate application program that acts as both a client and a server"); EX1057, 3 (describing that a "proxy" is "a device or program empowered to act for another").

d. 10[d]: *"accept, from the application, commands associated with the media object handle descriptor"*

167. *Bennett* discloses that the MA 206 receives "requests to control the media stream, such as a PAUSE request to pause an active media stream, and a RESUME request to resume a paused media stream" ("***accept … commands***"). EX1041, ¶76; EX1055, ¶¶24, 29, 60, 73 (describing other examples of commands). A POSITA would have recognized and found obvious that these commands are issued "***from the application***" because *Bennett* describes, in the same paragraph discussing the above commands, that "the user application 150 can direct how media or messages are routed." EX1041, ¶77. Additionally, in my opinion, the application is the interface between the user and the stream and, given that streaming commands like "pause" and "stop" are typically in response to user action, these commands would have typically originated "***from the application***" as a result of the user

interacting with the application. *See, e.g.*, EX1041, ¶¶24 (describing a "user application 150"), 77.

168. Additionally, the "PAUSE" and "RESUME" commands of *Bennett* are "***associated with the media object handle descriptor***" at least because these commands are applied to the stream described by the "***media object handle descriptor***." EX1041, ¶¶34-35, 50-56, 86, Table-3; EX1055, ¶¶8-9, 44-46, 51-54, 59, 71, FIG. 2; *see supra* §X.D.3.b (10[b]) (describing that a file track handle is created and shared for a stream once the stream is requested). Indeed, *Srikantan* discloses that, once a handle is created, each handle "includes methods to start, stop, pause and otherwise control a media stream (e.g., in response to client commands)" (such that these "methods" are "***associated with the media object handle descriptor***"). EX1055, ¶73, FIG. 5.

e. 10[e]: *"control playback of the media data by the media player based on the commands"*

169. *Bennett*'s PAUSE and RESUME commands control the media stream played by the MA 206's media player, and thus the MA 206 "***control[s] playback of the media data by the media player based on the commands***." EX1041, ¶¶77, 80, Table-3. *Srikantan* similarly describes that commands are applied to retrieved streams to manipulate the content. EX1055, ¶¶24, 29, 60, 73.

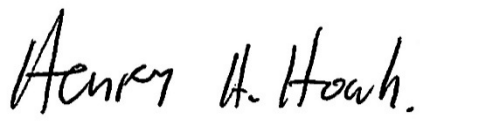
XI. CONCLUSION

170. For the reasons set forth above, it is my opinion that claims 1-19 of the '918 patent are unpatentable in view of the prior art. In signing this declaration, I understand that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I acknowledge that I may be subject to cross-examination in this 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.

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

Date: January 9, 2026

Respectfully submitted,



Dr. Henry Houh