

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

PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

I.	Introduction.....	1
II.	The '918 Patent.....	1
	A. Overview	1
	B. Prosecution History	4
	C. Critical Date of the '918 Patent.....	4
III.	Level of Ordinary Skill.....	7
IV.	Claim Construction.....	7
V.	Statement of Precise Relief Requested.....	8
VI.	The Challenged Claims Are Unpatentable	9
	A. Ground 1: <i>Bennett</i> and <i>Vadde</i> render Claims 1-3, 8-9, 13-14, and 19 Obvious.....	9
	1. Overview of <i>Bennett</i>	9
	2. Overview of <i>Vadde</i>	10
	3. Motivation to Combine <i>Bennett</i> and <i>Vadde</i> with Reasonable Expectation of Success	11
	4. Claim 1	16
	a. 1[pre]: “ <i>A wireless end-user device, comprising</i> ”	16
	b. 1[a]: “ <i>a wireless modem ...</i> ”	20
	c. 1[b]: “ <i>a network stack ...</i> ”	21
	d. 1[c]: “ <i>a first network stack Application Programming Interface (API) ...</i> ”	24
	e. 1[d]: “ <i>a second API ...</i> ”	38

f.	1[e]: “ <i>a media service manager ...</i> ”	44
g.	1[f]: “ <i>one or more service classification and measurement agents ...</i> ”	47
5.	Claim 2	56
6.	Claim 3	57
7.	Claim 8	58
8.	Claim 9	59
9.	Claim 13	62
10.	Claims 14 and 19	64
B.	Ground 2: Claims 4-6, 11, and 15-17 Are Obvious over <i>Bennett</i> in View of <i>Vadde</i> and <i>Riggs</i>	65
1.	Overview of <i>Riggs</i>	65
2.	Motivation to Combine <i>Bennett-Vadde</i> and <i>Riggs</i> with Reasonable Expectation of Success	66
3.	Claim 4	70
4.	Claim 5	72
5.	Claim 6	74
6.	Claim 11	77
a.	11[a]	77
b.	11[b]	78
c.	11[c]	80
7.	Claims 15-17	82

C.	Ground 3: Claims 7, 12, and 18 Are Obvious over <i>Bennett</i> in View of <i>Vadde</i> , <i>Riggs</i> , and <i>Hendrickson</i>	83
1.	Overview of <i>Hendrickson</i>	83
2.	Motivation to Combine <i>Bennett-Vadde-Riggs</i> and <i>Hendrickson</i> with Reasonable Expectation of Success	84
3.	Claim 7	87
4.	Claim 12	90
a.	12[a]	90
b.	12[b]	92
c.	12[c]	92
5.	Claim 18	94
D.	Ground 4: Claim 10 Is Obvious over <i>Bennett</i> in View of <i>Vadde</i> , <i>Riggs</i> , and <i>Srikantan</i>	95
1.	Overview of <i>Srikantan</i>	95
2.	Motivation to Combine <i>Bennett-Vadde-Riggs</i> and <i>Srikantan</i> with Reasonable Expectation of Success	97
3.	Claim 10	99
a.	10[a]	99
b.	10[b]	99
c.	10[c]	101
d.	10[d]	102
e.	10[e]	103

VII.	Mandatory Notices.....	104
A.	Real Parties-in-Interest.....	104
B.	Related Matters.....	106
C.	Lead and Back-Up Counsel.....	107
D.	Service Information.....	112
VIII.	Grounds for Standing.....	112
IX.	Conclusion	112

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Accent Packaging, Inc. v. Leggett & Platt, Inc.</i> , 707 F.3d 1318 (Fed. Cir. 2013)	36-37
<i>CUPP Computing AS v. Trend Micro Inc.</i> , 53 F.4th 1376 (Fed. Cir. 2022)	36
<i>Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015)	4
<i>In re Harza</i> , 274 F.2d 669 (CCPA 1960)	77-78
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	17, 70, 86, 99
<i>Liebel-Flarsheim Co. v. Medrad, Inc.</i> , 358 F.3d 898 (Fed. Cir. 2004)	37
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	7, 37
<i>Renishaw PLC v. Marposs Societa’ per Azioni</i> , 158 F.3d 1243 (Fed. Cir. 1998)	37
<i>Samsung Elecs. Co. Ltd. v. Headwater Rsch. LLC</i> , IPR2024-01396, Paper 13 (P.T.A.B. Apr. 1, 2025)	1, 7, 8, 38
Statutes	
35 U.S.C. § 311	8

Rules

37 C.F.R. § 42.6	115
37 C.F.R. § 42.24	114
37 C.F.R. § 42.100(b)	7

LISTING OF CLAIMS

1[pre]	A wireless end-user device, comprising
1[a]	a wireless modem configurable to connect to a wireless network;
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 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;
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;
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
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.
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.
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.
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.
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.
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.
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.
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.
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,
10[b]	return to the application a media object handle descriptor,
10[c]	call a proxy service to perform one or more network data transfers comprising the media object,
10[d]	accept, from the application, commands associated with the media object handle descriptor, and
10[e]	control playback of the media data by the media player based on the commands.

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;
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
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.
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;
12[b]	a user interface; and
12[c]	a user interface display agent to display the data usage classified by application to a user.
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.

14[pre]	14. A method of operating a wireless end-user device when connected via a wireless modem to a wireless network, the method comprising:
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 network stack coupled to the wireless modem;
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;
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
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.
15[pre]	A wireless end-user device, comprising:
15[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;
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;
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;
15[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
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.

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.
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.
18	The wireless end-user device of claim 17, further comprising a usage and classification reconciliation agent and a 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.
19[pre]	A wireless end-user device, comprising:
19[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;
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;
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 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 through the device networking stack; and
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.

LIST OF EXHIBITS

<u>Exhibit</u>	<u>Description</u>
EX1001	U.S. Patent No. 9,647,918 to Gregory G. Raleigh, et al. (“the ’918 patent”).
EX1002	Prosecution History of the ’918 patent (“the Prosecution History”).
EX1003	Declaration of Henry Houh, Ph.D.
EX1004	Curriculum Vitae of Henry Houh, Ph.D.
EX1005	Infringement Contentions, <i>Headwater Research LLC v. Samsung Electronics Co., Ltd. et al.</i> , 2-23-cv-00641 (EDTX), filed May 15, 2024.
EX1006	U.S. Provisional Application No. 61/264,126
EX1007	U.S. Provisional Application No. 61/270,353
EX1008	U.S. Provisional Application No. 61/275,208
EX1009	U.S. Provisional Application No. 61/237,753
EX1010	U.S. Provisional Application No. 61/264,120
EX1011	U.S. Provisional Application No. 61/348,022
EX1012	U.S. Provisional Application No. 61/381,159
EX1013	U.S. Provisional Application No. 61/381,162
EX1014	U.S. Provisional Application No. 61/384,456
EX1015	U.S. Provisional Application No. 61/389,547
EX1016	U.S. Provisional Application No. 61/385,020
EX1017	U.S. Provisional Application No. 61/387,243
EX1018	U.S. Provisional Application No. 61/387,247

<u>Exhibit</u>	<u>Description</u>
EX1019	U.S. Provisional Application No. 61/407,358
EX1020	U.S. Provisional Application No. 61/418,507
EX1021	U.S. Provisional Application No. 61/418,509
EX1022	U.S. Provisional Application No. 61/420,727
EX1023	U.S. Provisional Application No. 61/422,565
EX1024	U.S. Provisional Application No. 61/422,572
EX1025	U.S. Provisional Application No. 61/422,574
EX1026	U.S. Provisional Application No. 61/252,151
EX1027	U.S. Provisional Application No. 61/252,153
EX1028	U.S. Provisional Application No. 61/206,354
EX1029	U.S. Provisional Application No. 61/206,944
EX1030	U.S. Provisional Application No. 61/207,393
EX1031	U.S. Provisional Application No. 61/207,739
EX1032	U.S. Application No. 12/695,019
EX1033	U.S. Application No. 12/695,020
EX1034	U.S. Application No. 12/694,445
EX1035	U.S. Application No. 12/694,451
EX1036	U.S. Application No. 12/694,455
EX1037	U.S. Application No. 12/695,980
EX1038	U.S. Application No. 12/695,021
EX1039	U.S. Application No. 12/380,780

<u>Exhibit</u>	<u>Description</u>
EX1040	U.S. Application No. 12/380,778
EX1041	U.S. Patent Publication No 2006/0149811 (“ <i>Bennett</i> ”)
EX1042	U.S. Patent Publication No. 2012/0117478 (“ <i>Vadde</i> ”)
EX1043	U.S. Patent No. 8,429,516 (“ <i>Riggs</i> ”)
EX1044	EP Patent Publication No. 1 850 575 A1 (“ <i>Rybak</i> ”)
EX1045	U.S. Patent Publication No. 2004/0260630 (“ <i>Benco</i> ”)
EX1046	U.S. Patent No. 6,578,077 (“ <i>Rakoshitz</i> ”)
EX1047	U.S. Patent Publication No. 2006/0223495 (“ <i>Cassett</i> ”)
EX1048	U.S. Patent Publication No. 2008/0080458 (“ <i>Cole</i> ”)
EX1049	U.S. Patent Publication No. 2008/0209451 (“ <i>Michels</i> ”)
EX1050	U.S. Patent Publication No. 2006/0039354 (“ <i>Rao</i> ”)
EX1051	RESERVED
EX1052	RESERVED
EX1053	U.S. Provisional Application No. 61/435,564
EX1054	U.S. Patent No. 6,754,470 (“ <i>Hendrickson</i> ”)
EX1055	U.S. Patent Publication No. 2002/0056126 (“ <i>Srikantan</i> ”)
EX1056	Newton’s Telecom Dictionary, 24th Edition
EX1057	Webster’s New World, Telecom Dictionary
EX1058	Wiley Electrical and Electronics Engineering (IEEE) Dictionary
EX1059	The Authoritative Dictionary of IEEE Standards Terms
EX1060	Claim Construction Order, <i>Headwater Rsch. LLC v. Samsung Elecs. Co. Ltd.</i> , 2:23-cv-00641-JRG-RSP (E.D. Tex. Apr. 7, 2025).

<u>Exhibit</u>	<u>Description</u>
EX1061	J. Postel, Transmission Control Protocol, RFC 793, Internet Engineering Task Force (Sept. 1981) (“RFC 793”)
EX1062	U.S. Patent Publication No. 2008/0122796 (“ <i>Jobs</i> ”)
EX1063	U.S. Patent Publication No. 2010/0017506 (“ <i>Fadell</i> ”)
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”)

I. Introduction

Google LLC (“Petitioner”) requests institution of *inter partes* review and cancellation of claims 1-19 of U.S. Patent No. 9,647,918. The Board previously instituted *inter partes* review of the ’918 patent on the same grounds presented in this Petition. *Samsung Elecs. Co. Ltd. v. Headwater Rsch. LLC*, IPR2024-01396, Paper 13, 1, 15 (P.T.A.B. Apr. 1, 2025). The Board should reach the same conclusion here and institute review.

II. The ’918 Patent

A. Overview

The ’918 patent is directed to “a wireless end-user device” that includes a “proxy network service manager” that facilitates media requests from applications. EX1001, Abstract, 71:21-42, 110:12-111:17, 119:49-60, FIGS. 30, 35. In an embodiment, 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. Another embodiment depicts application 3202 communicating through a “control application 3204” positioned between the application and proxy service manager 3206. *Id.*, 116:39-118:50, FIG. 32. Yet another embodiment 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. EX1003, ¶53.

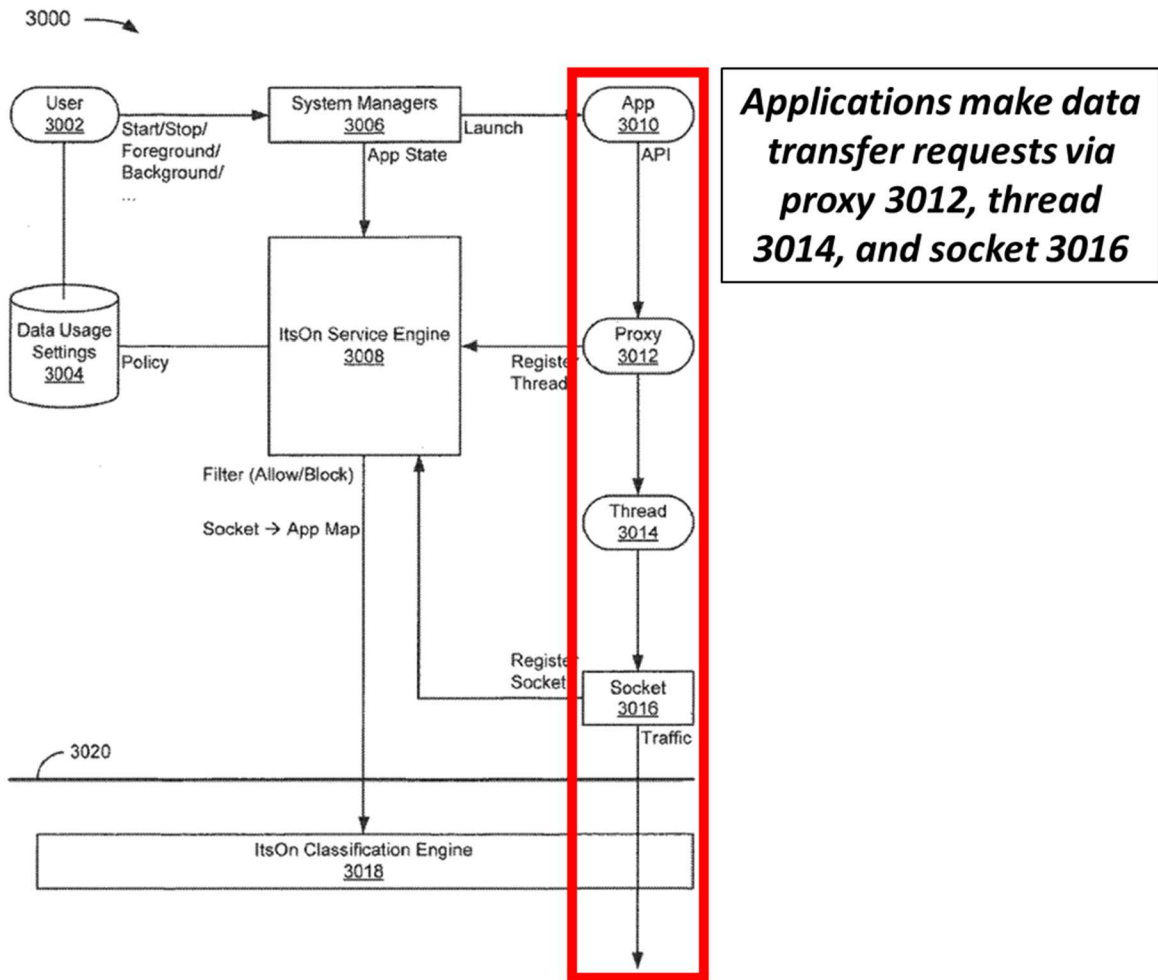


FIG. 30

EX1001, FIG. 30.¹

¹ Annotations to figures are shown in color.

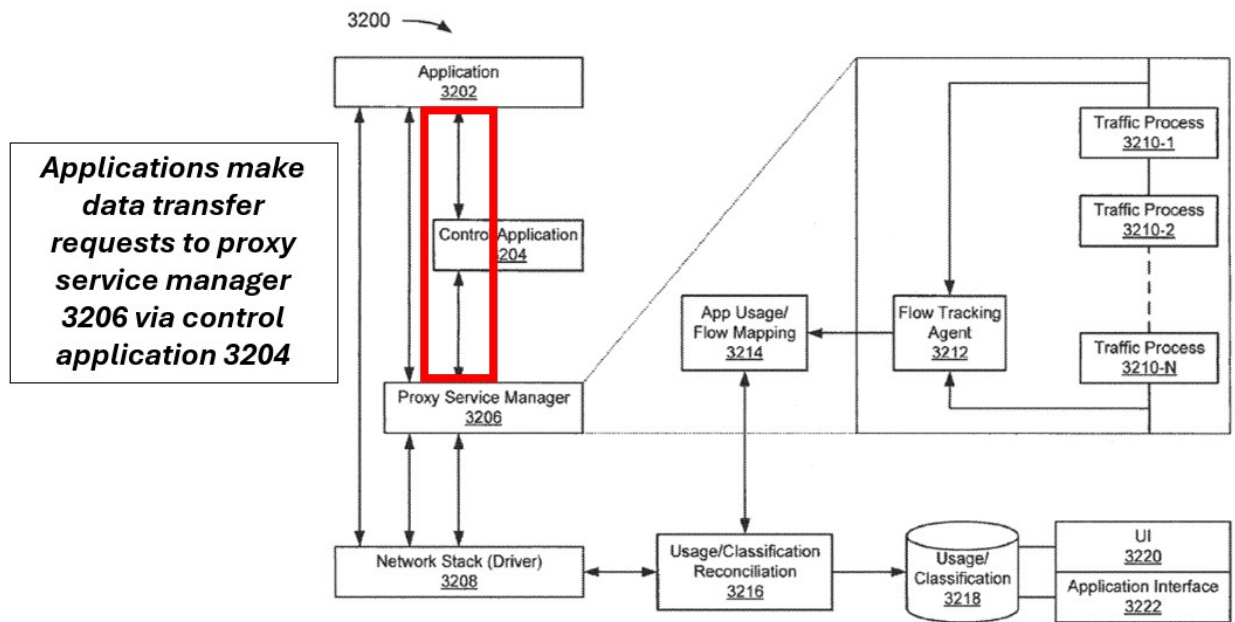


FIG. 32

EX1001, FIG. 32.

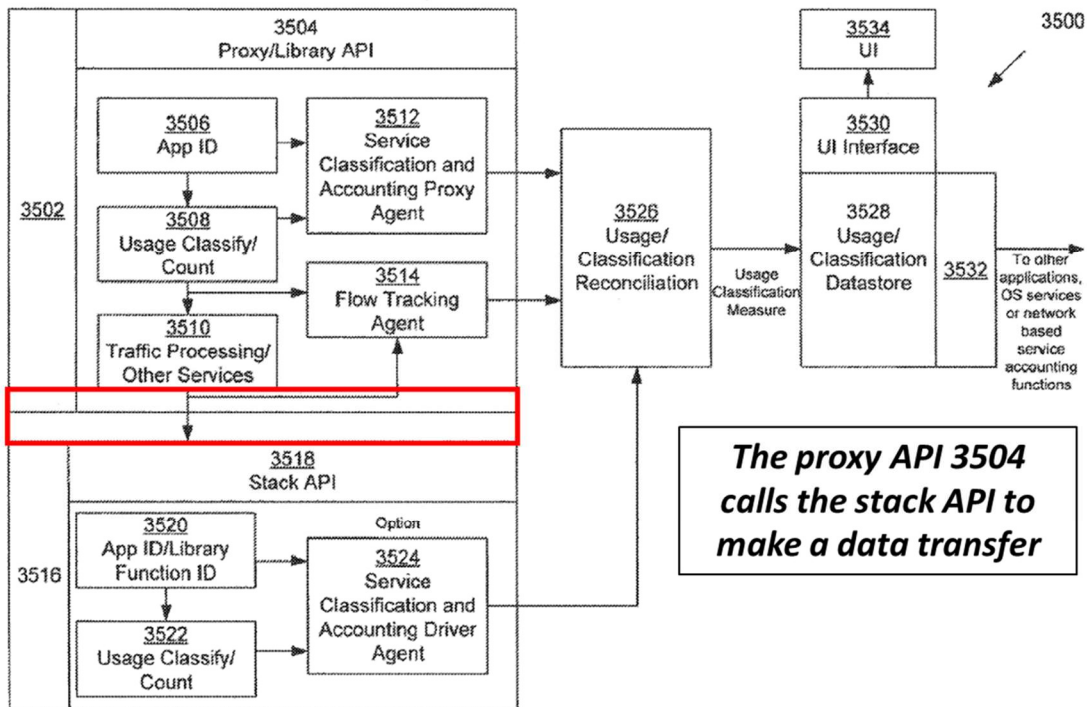


FIG. 35

EX1001, FIG. 35.

B. Prosecution History

The Examiner issued one office action during prosecution. EX1002, 381-390. The applicant amended the claims and added new claims 17-21 (issued claims 15-19). EX1002, 99-112. All claims were then allowed. EX1002, 21-25; EX1003, ¶54.

C. Critical Date of the '918 Patent

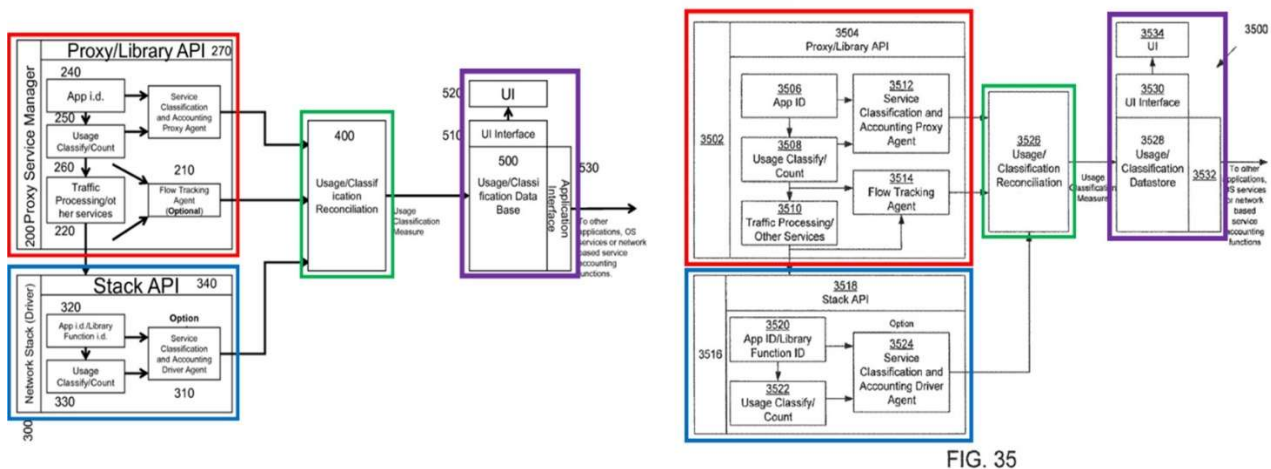
Patent Owner has advanced a Critical Date of January 24, 2011 for the '918 patent in prior litigation. EX1005, 11-12. Patent Owner also stipulated before the Board that it would not assert an earlier priority claim to disqualify the *Vadde* reference (EX1042) asserted in this Petition. *Samsung*, IPR2024-01397, Paper 13 at 5-6. Based on Patent Owner's assertions and stipulations, a Critical Date of January 24, 2011 should be used for this proceeding. The following analysis confirms this is the correct date.

None of the Challenged Claims are entitled to the filing date before January 24, 2011, because there is no prior written description support in any earlier application. See EX1006-EX1040; see *Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378-80 (Fed. Cir. 2015). Claims 1, 14, 15, and 19 recite a "first network stack ... (API)," a "second API," and "a media service manager ... to manage network data transfers for [a] media object." Claims 5-7 and 16-18 recite a "requesting application storing function." These features appear

nowhere in the pre-January 24, 2011 provisionals or pre-January 24, 2011 applications to which the '918 patent claims priority. EX1006-1040. Because all claims depend on these features, **none** of the claims are entitled an earlier date. EX1001, Claims 1-19.

At most, the claims of the '918 patent are entitled to the filing date of **January 24, 2011** based on provisional application 61/435,564 (“the '564 provisional,” EX1053) where these features first appear—consistent with what Patent Owner has proposed in the co-pending litigation. EX1005, 11-12; *see also* EX1006-1040. For example, the '564 provisional was the first application to add disclosure relating to FIG. 35 of the '918 patent, including a description of a “proxy service manager” (a “media service manager”). EX1053, 8, 15.

Page 15 of the '564 provisional (left) appears to depict FIG. 35 of the '918 patent (right)



EX1053, 21 (left) compared to EX1001, FIG.35 (right).

Proxy Service Manager



- In some embodiments, a proxy network service manager refers to an intermediary data flow function in a device operating system that sits on a data path between a device application and a device networking stack interface to provide a level of network service abstraction from the network stack interface, a higher level service function above the network stack interface, enhanced or special traffic processing functions, media service transfer management, file download service, HTTP proxy service functions, QoS differentiation, or other similar or related higher level traffic processing.
- Example Proxy Service Managers include the following: **media service manager** (e.g. android media service library function), email service manger, DNS function, software download service manager, media download manager, data download service manager, Android "media" library function, Android.net library function, Jave.net library function, Apache library function, other similar software/library functions or services in other device operating systems, SMTP/IMAP/POP proxy, HTTP proxy, IM proxy, VPN service manager, SSL proxy, etc.

The '564 provisional introduced the "media service manager"

EX1053, 8.

The references below predate January 24, 2011, and thus constitute prior art under pre-AIA 35 U.S.C. §§ 102(a), (b), and/or (c):

Reference	Filing Date	Publication Date
US 2006/0149811 (" <i>Bennett</i> ") EX1041	4/26/2005	7/6/2006
US 2012/0117478 (" <i>Vadde</i> ") EX1042	11/8/2010	5/10/2012
US 8,429,516 (" <i>Riggs</i> ") EX1043	8/20/2007	4/23/2013
US 6,754,470 (" <i>Hendrickson</i> ")	8/31/2001	6/22/2004

Reference	Filing Date	Publication Date
EX1054		
US 2002/0056126 (“ <i>Srikantan</i> ”) EX1055	4/6/2001	5/9/2002

III. Level of Ordinary Skill

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. EX1003 ¶¶59-60. More related education could compensate for fewer years of work experience (and vice versa). *Id.*

IV. Claim Construction

Claims in an IPR are construed under *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). 37 C.F.R. § 42.100(b). In IPR2024-01396, Patent Owner 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. IPR2024-01396, Paper 13, 13-15. Headwater did not seek this narrow construction in the district court. *See* EX1060, 1-5. Nor did the district court enter one. *See id.* As discussed below, the Board should decline to narrowly

construe the claims to require direct access between the claimed applications and the APIs. Instead, the plain meaning of the claims, supported by embodiments in the specification, permits indirect access between the applications and API. *See* §§VI.A.4.d (1[c]), VI.A.4.e (1[d]). Indeed, the Board previously agreed that the claims encompass indirect access. *Samsung*, IPR2024-01396, Paper 13 at 13-15.

No other terms need to be expressly construed, consistent with Headwater's position in district court litigation. *See* EX1060, 1-5. Petitioner reserves the right to respond to any constructions offered by Patent Owner or adopted by the Board.

V. Statement of Precise Relief Requested

Petitioner requests review under 35 U.S.C. § 311 of claims 1-19 of the '918 patent and cancellation of those claims based on the following grounds.

Ground	References	Challenged Claims
1	<i>Bennett</i> in view of <i>Vadde</i>	1-3, 8-9, 13-14, and 19
2	<i>Bennett</i> in view of <i>Vadde</i> and <i>Riggs</i>	4-6, 11, and 15-17
3	<i>Bennett</i> in view of <i>Vadde</i> , <i>Riggs</i> and <i>Hendrickson</i>	7, 12 and 18
4	<i>Bennett</i> in view of <i>Vadde</i> , <i>Riggs</i> and <i>Srikantan</i>	10

VI. The Challenged Claims Are Unpatentable

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

1. Overview of *Bennett*

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. *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 media using “a local media rendering device.” EX1041, ¶¶25, 76, FIG. 3. The media can also be routed directly to the application. ¶76, FIG. 10; EX1003, ¶61.

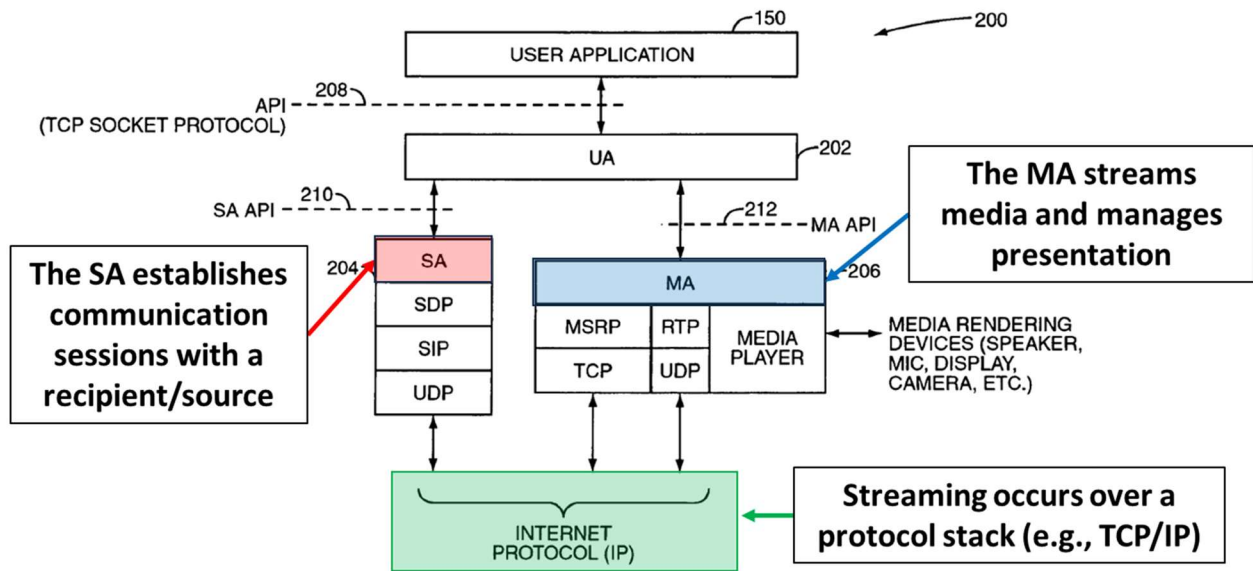


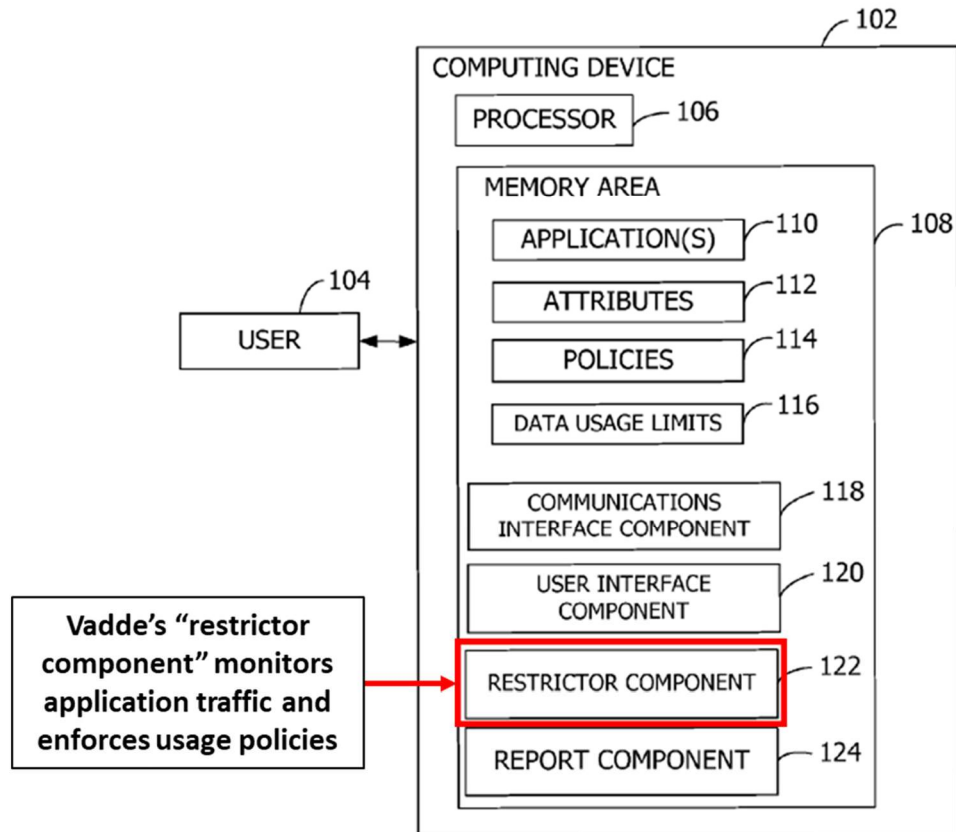
FIG. 3

EX1041, FIG. 3.

2. Overview of *Vadde*

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.” EX1042, ¶¶16, 22; EX1003, ¶62.

FIG. 1



EX1042, FIG. 1.

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

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. A POSITA would have been motivated to make this combination for multiple reasons. EX1003, ¶63.

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.” EX1003, ¶64; EX1042, ¶1. 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; EX1003, ¶64.

Indeed, the benefits of data usage monitoring were well-known prior to the ’918 patent and would have been known to a POSITA. EX1003, ¶65. 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. *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. EX1003, ¶65.

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). EX1003, ¶66; 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. EX1003, ¶66. 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); EX1003, ¶66.

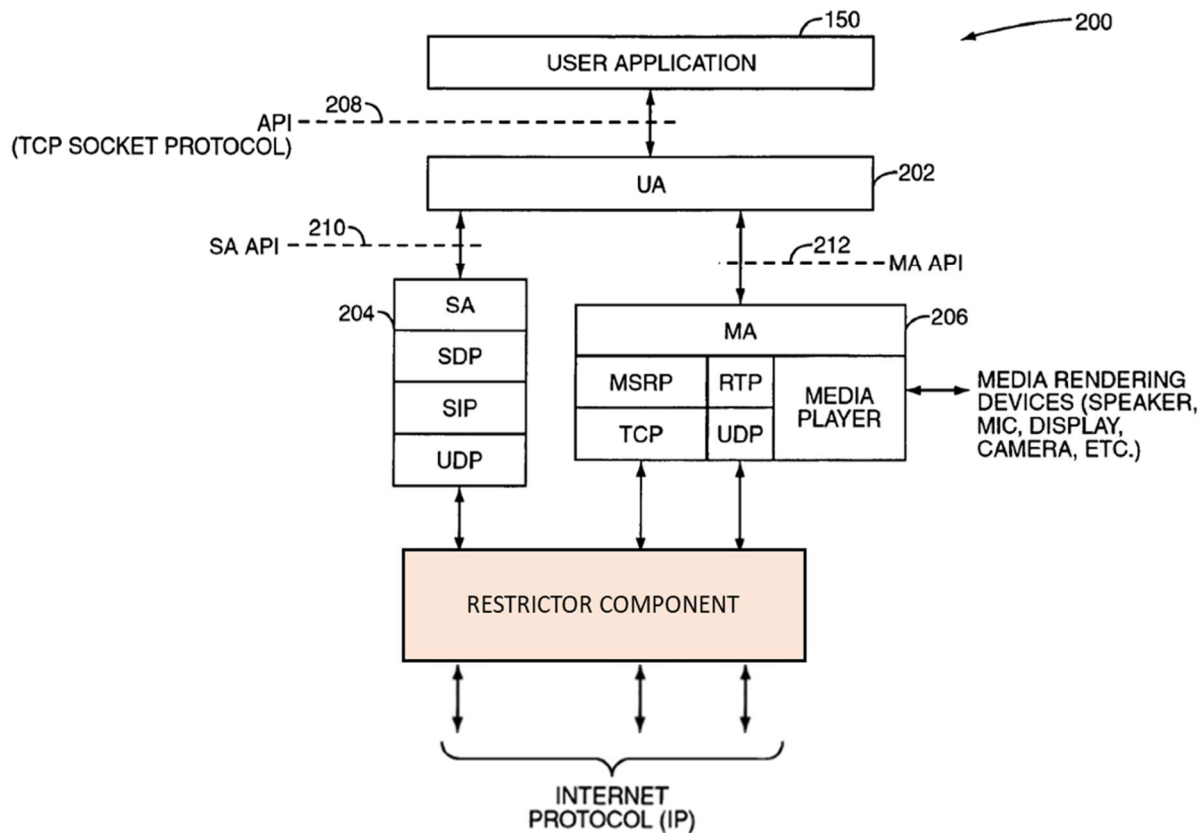
Finally, adding per-application data usage monitoring to the *Bennett* device would have enabled device manufacturers and service providers to incorporate additional applications and functionality into wireless devices while allowing users to retain control of aggregate device data usage. EX1003, ¶67; 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; EX1003, ¶67.

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). *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 417 (2007). 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. EX1003, ¶68.

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. EX1003, ¶69. 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; EX1003, ¶69, 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; EX1003, ¶69. 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* corroborates this and discloses 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. *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.



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

4. Claim 1

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

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, ... , *cameras*, *music/media players*, ... , and any other networked device.” EX1001, 40:28-35²; EX1003, ¶70.

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*”) that communicate over wireless networks). EX1041, ¶78; *see also* ¶¶2, 5, 25, FIGS. 1, 4, 11; EX1003, ¶71.

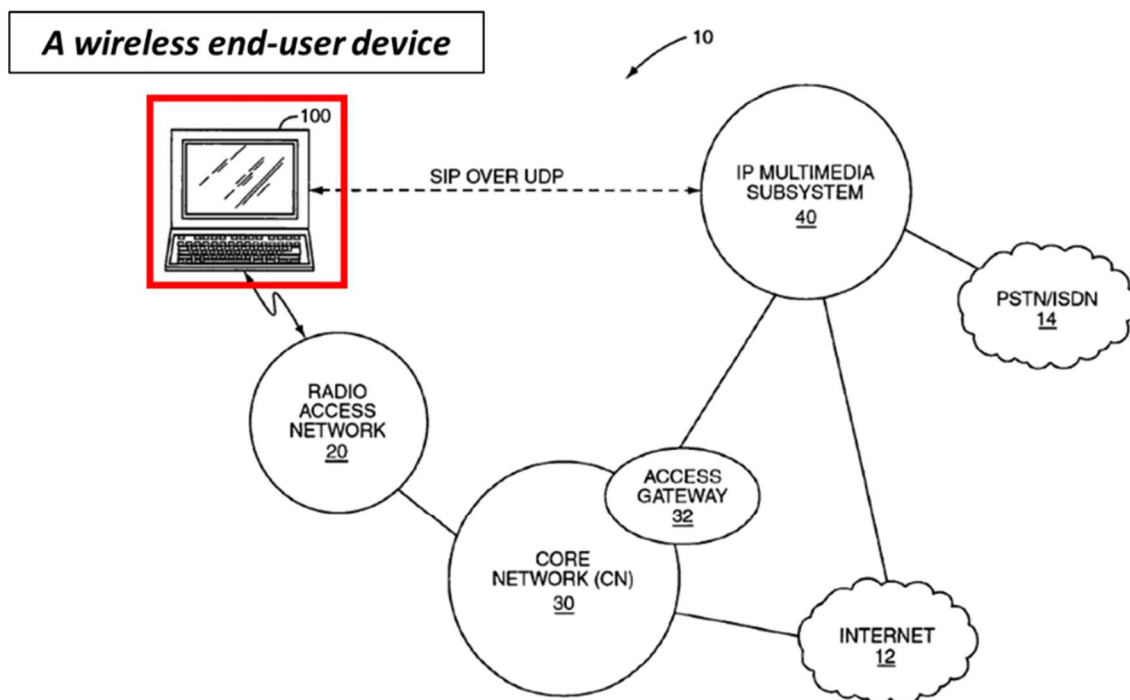


FIG. 1

EX1041, FIG. 1.

² All emphasis is added unless otherwise noted.

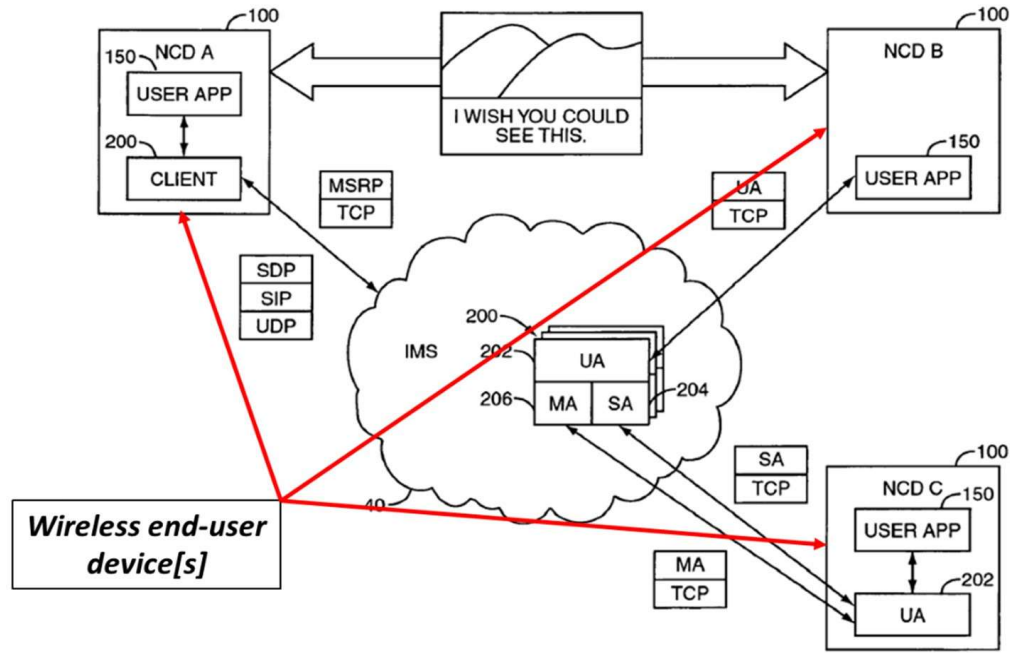


FIG. 4

EX1041, FIG. 4.

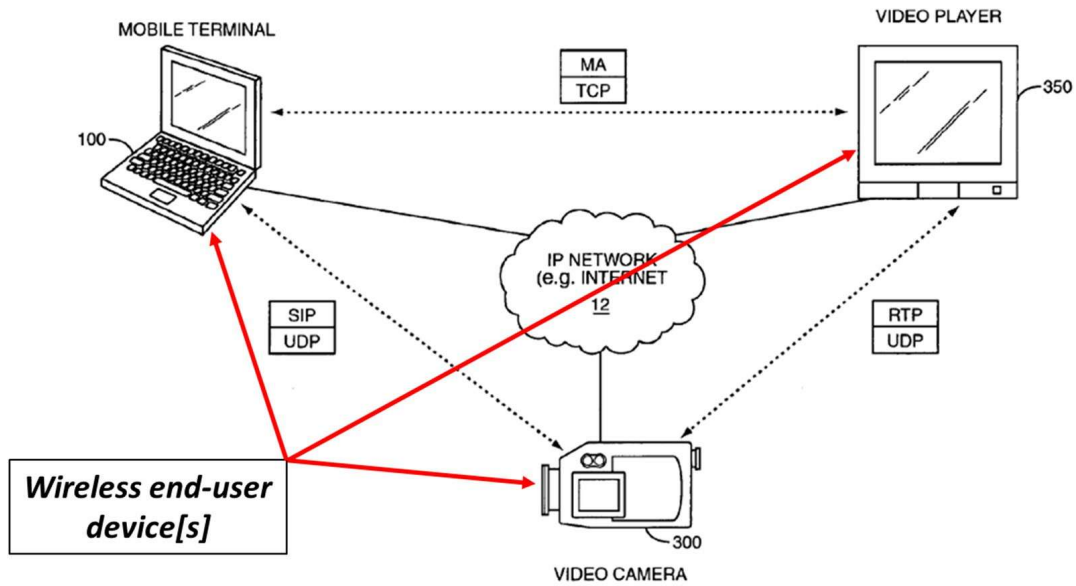
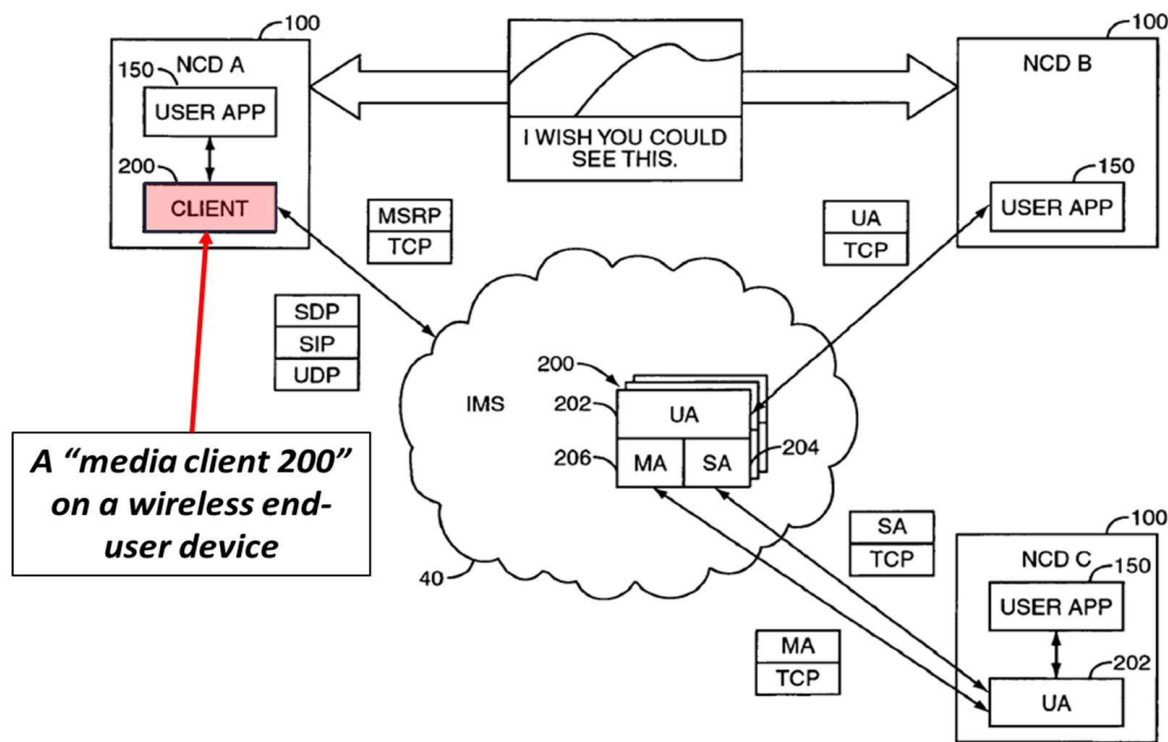


FIG. 11

EX1041, FIG. 11.

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 contained within a “*wireless end-user device*.” EX1041, ¶¶28, 78, FIGS. 4, 11; EX1003, ¶72.



EX1041, FIG. 4.

b. 1[a]: “a wireless modem ...”

Bennett discloses that its mobile devices operate on various “mobile communication network[s].” EX1041, ¶17, FIG. 1; EX1003, ¶73. *Bennett*’s networks include “General Packet Radio Services (GPRS) network[s]” and Universal Mobile Telecommunications Service (“UMTS”) networks (“**wireless network[s]**”). EX1041, ¶17.

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.**” EX1003, ¶74. 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. *Cassett* confirms that devices communicating in “GPRS” and “UMTS” networks included “wireless modems.” *Id.*

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.**” EX1048, ¶¶34-35, FIG. 2; EX1013, ¶¶125, 130; EX1003, ¶75. 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 ...*”

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, 8-12³; EX1003, ¶76.

As described above, communication over wireless networks on the *Bennett* mobile end-user device occurs via “*the wireless modem and the wireless network*.” See §VI.A.4.b. (1[a]); EX1003, ¶77.

³ *Bennett*’s figures all denote “media client 200,” thus indicating the same embodiment. EX1041, FIGS. 3-4, 8-12.

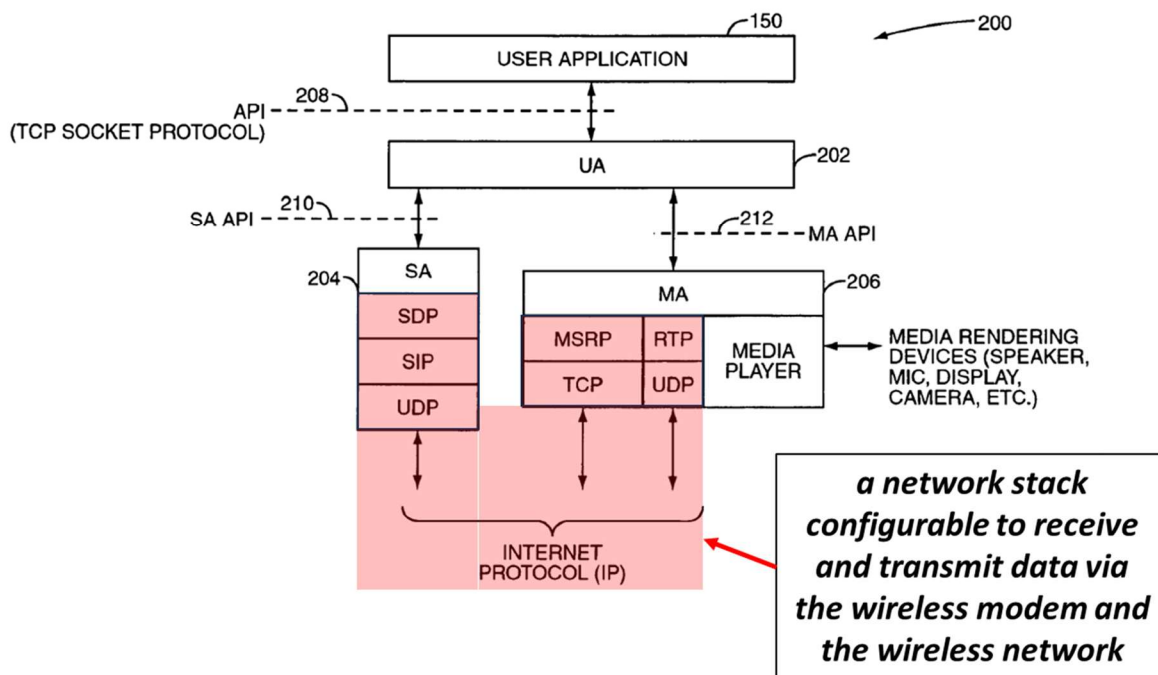


FIG. 3

EX1041, FIG. 3.

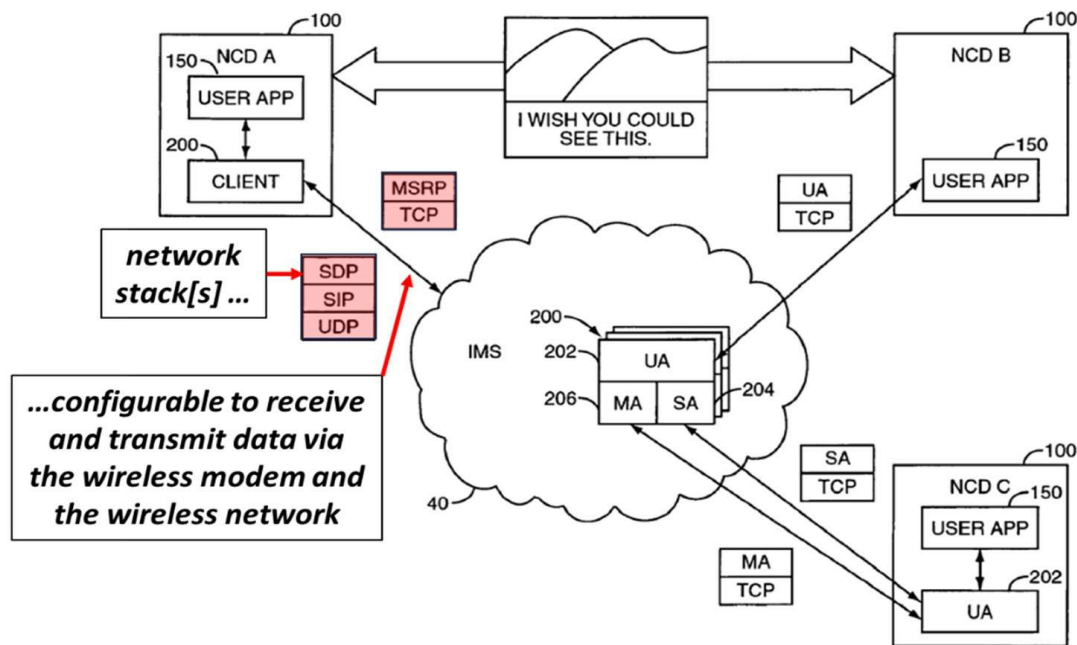
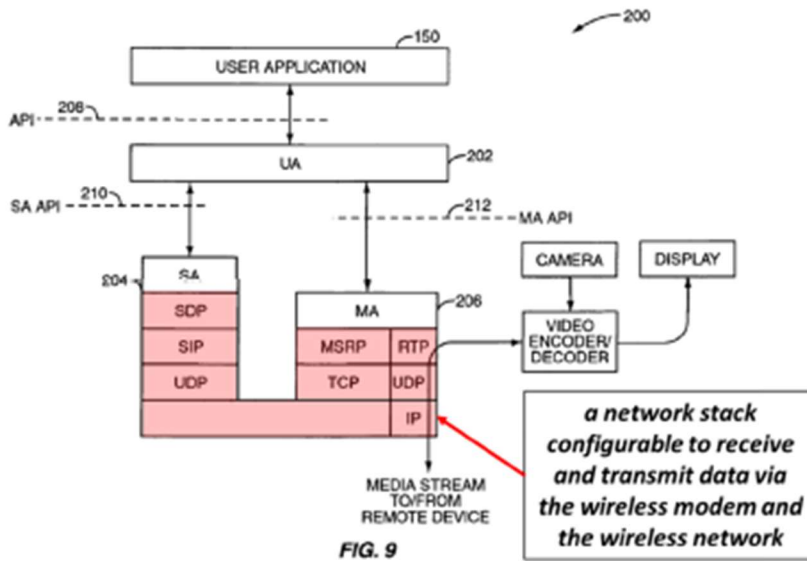
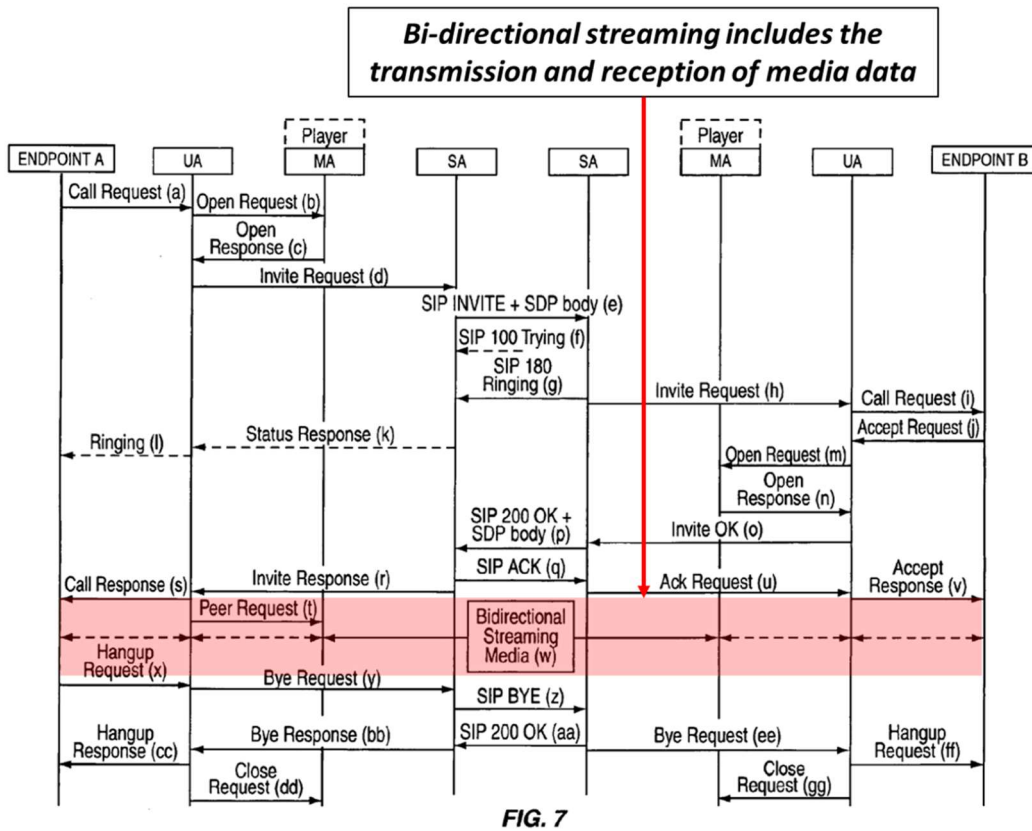


FIG. 4

EX1041, FIG. 4.

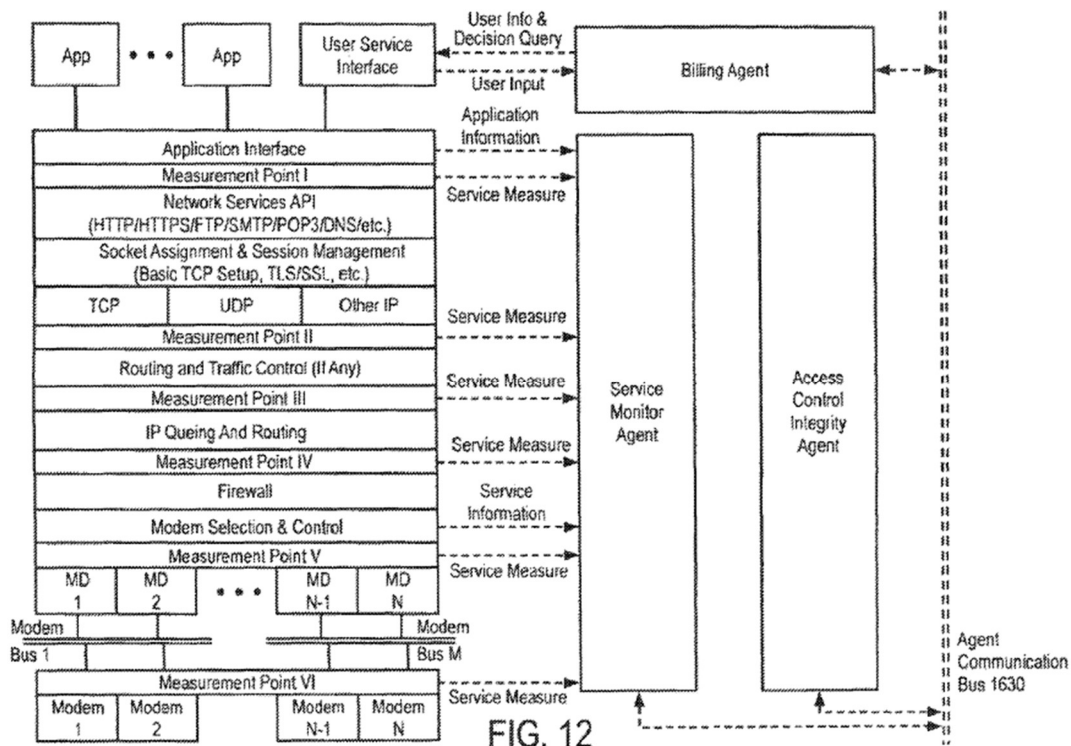


EX1041, FIG. 9.



EX1041, FIG. 7.

Similarly, the '918 patent depicts Internet Protocol ("IP"), Transmission Control Protocol ("TCP"), and User Datagram Protocol ("UDP") as part of a "device stack" (e.g., a "network stack"). EX1001, 2:21-26, FIGS. 12-13; EX1003, ¶78. 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.



EX1001, FIG.12

- d. 1[c]: *"a first network stack Application Programming Interface (API) ..."*

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

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; EX1003, ¶79. 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.*

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. EX1003, ¶80; 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. EX1003, ¶80; *see* §§VI.A.4.b. (1[a]), VI.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”

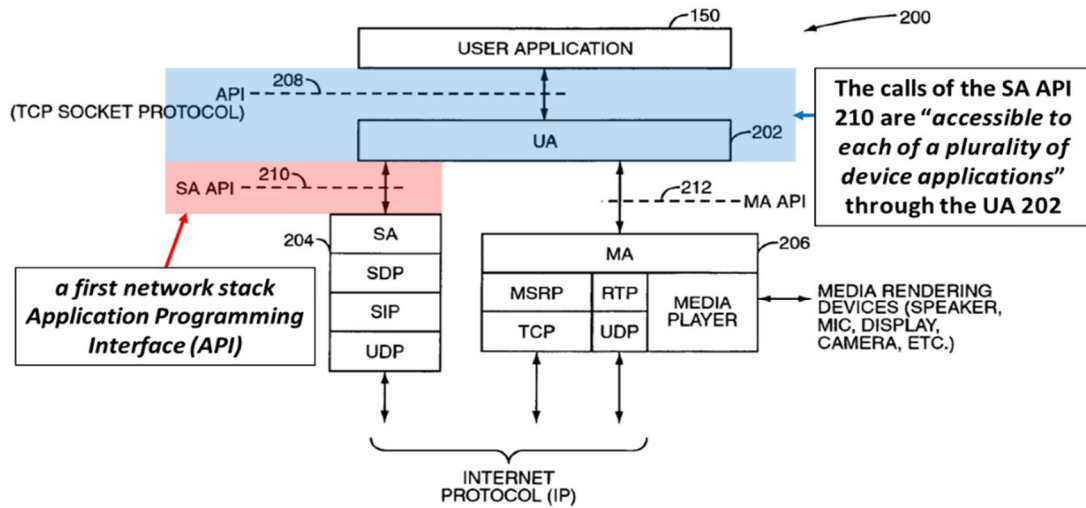
As discussed, *Bennett*’s signaling agent (SA) 204 within its “media client 200” that “implements SIP and SDP protocols to handle signaling tasks.” EX1041, ¶¶25, 31, 33, 40, 43-49, Table-2, FIGS. 3-10; EX1003, ¶81. The SA 204 is called by the “user agent (UA) 202” of the media client 200 using a “SA 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. SA API 210 is an API that provides access to the network stack (*see* §VI.A.4.c (1[b])) that provides access to a set of protocols enabling communication over a network. EX1003, ¶82. 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. EX1003, ¶82.

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.” *Id.*, ¶30. Requests from the user application via UA 202 “requiring action by ... the SA 204 ... start a transaction.” *Id.*, ¶31. In one exemplary request, 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); EX1003, ¶83. 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.*” EX1003, ¶83.

Bennett frequently refers to “applications” 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;
EX1003, ¶84.



EX1041, FIG. 3.

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

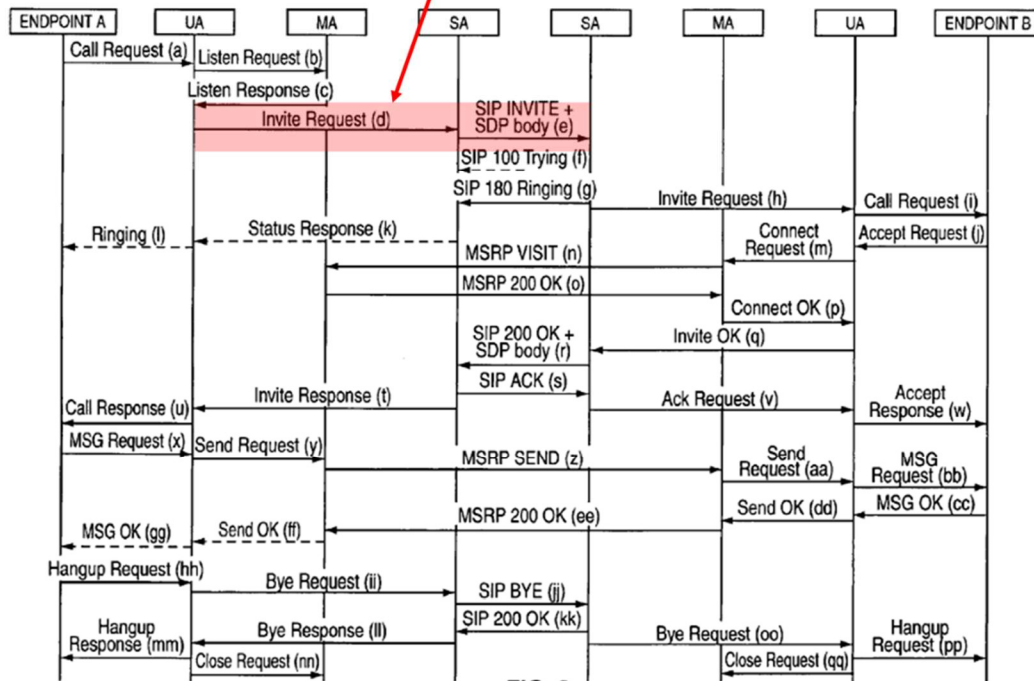
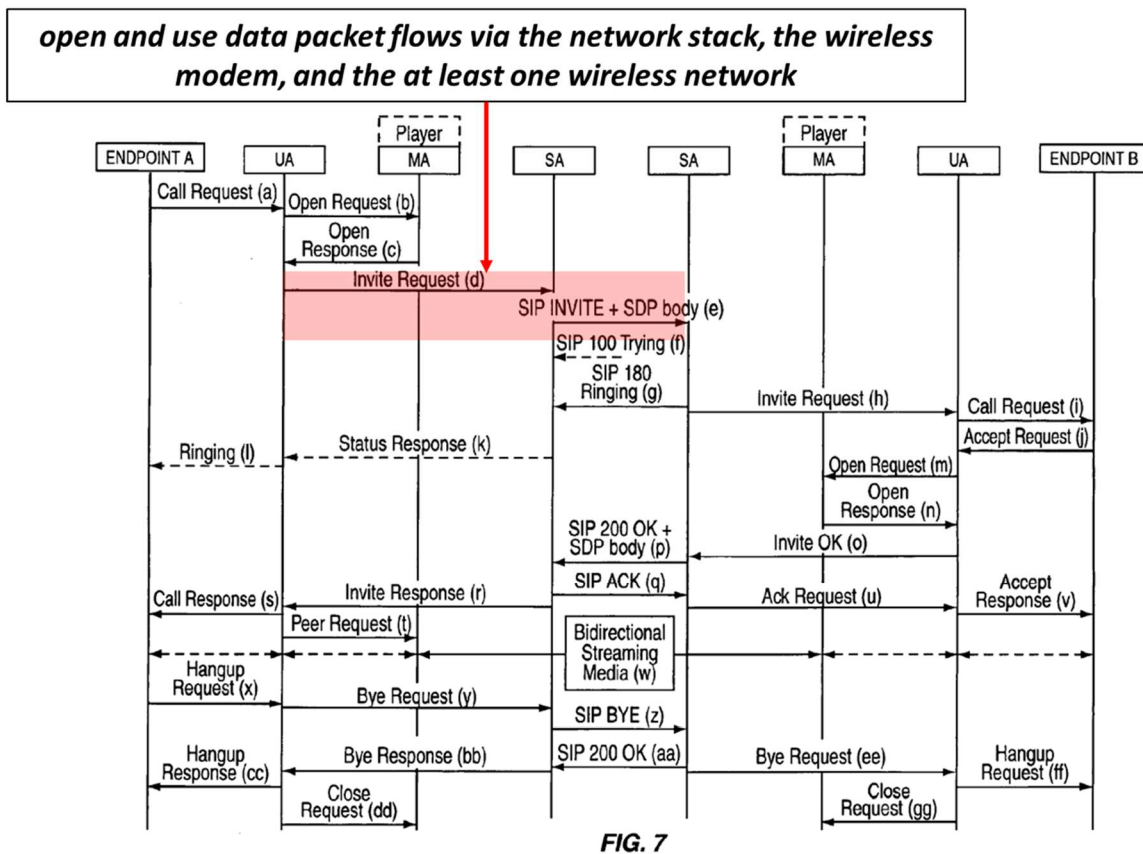


FIG. 6

EX1041, FIG. 6.



EX1041, FIG. 7.

To the extent Patent Owner argues 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. EX1003, ¶85.

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; EX1003, ¶86. 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, FIG. 30; EX1003, ¶86.

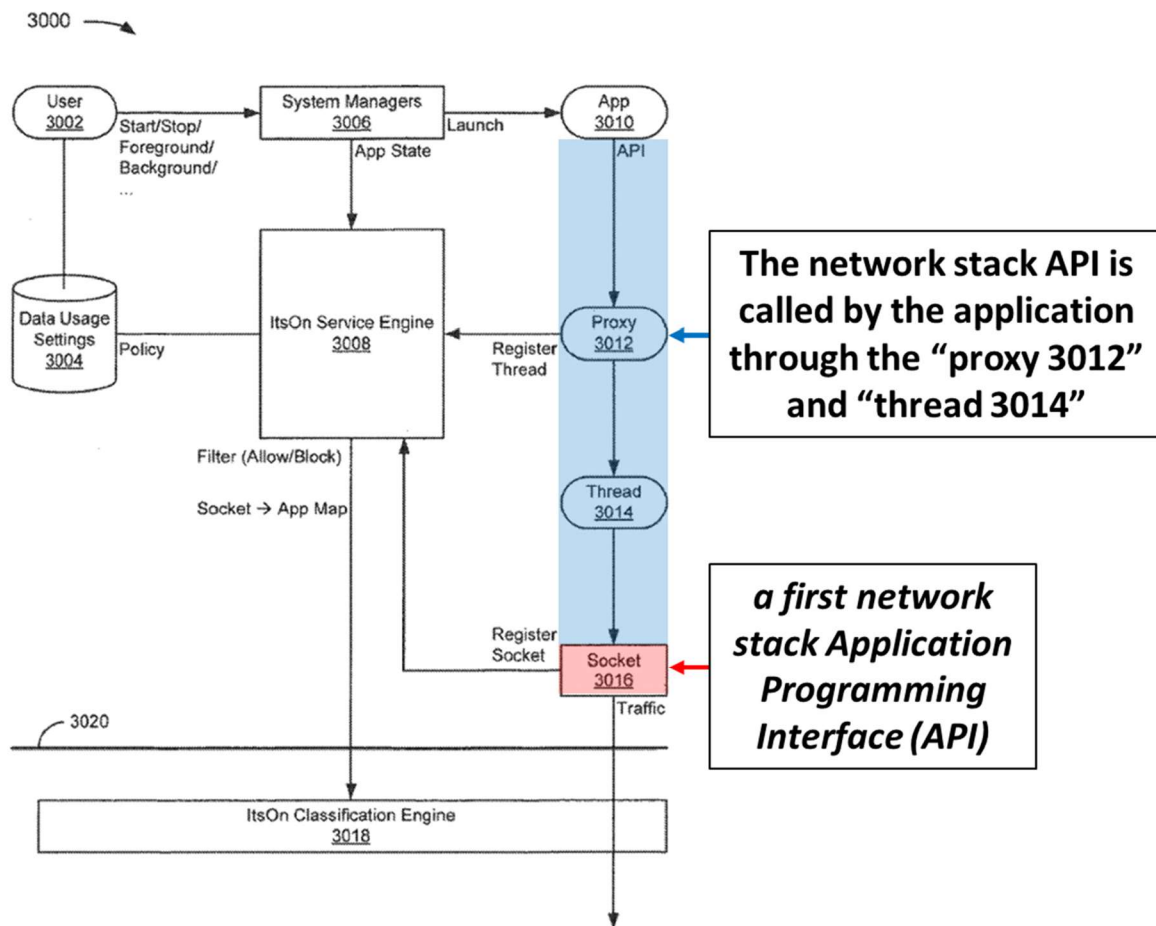


FIG. 30

EX1001, FIG. 30.

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; EX1003, ¶87. 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). EX1003, ¶87.

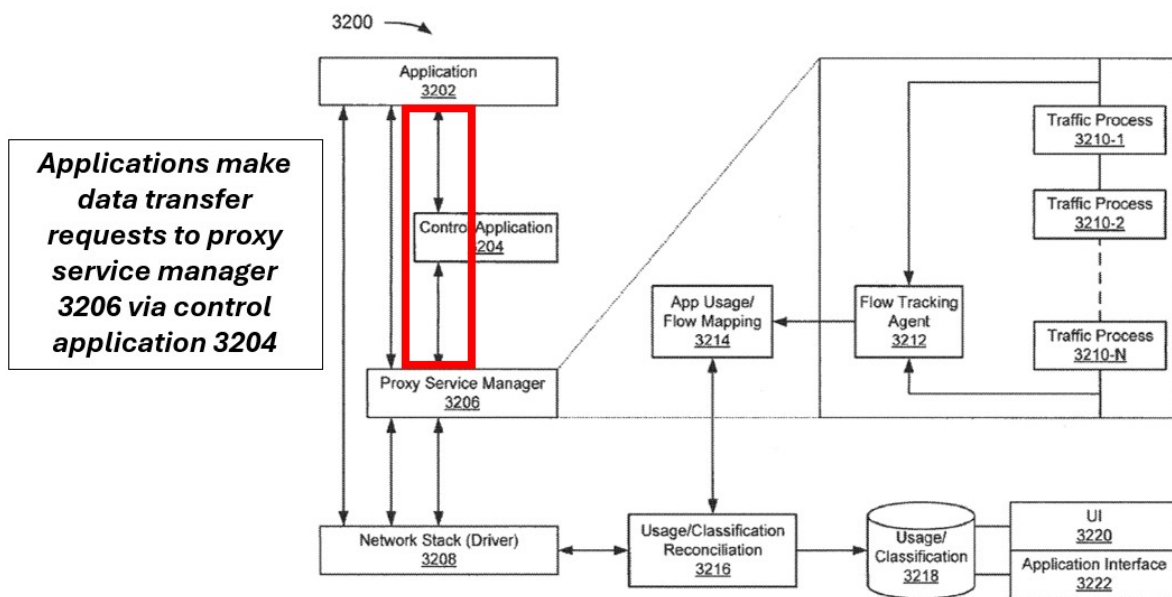
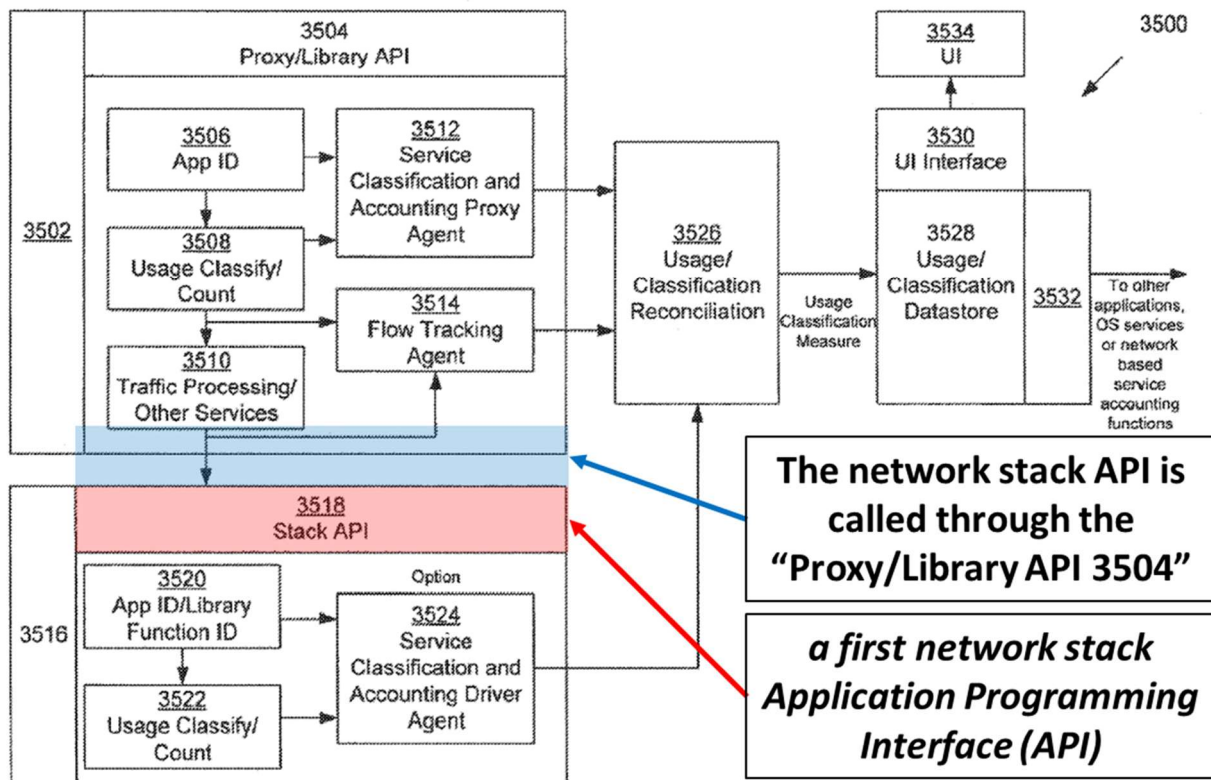


FIG. 32

EX1001, FIG. 32.



EX1001, FIG. 35.

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; EX1003, ¶88.

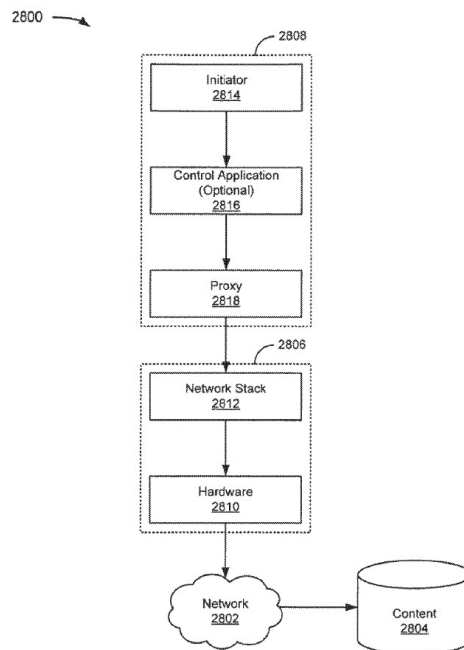
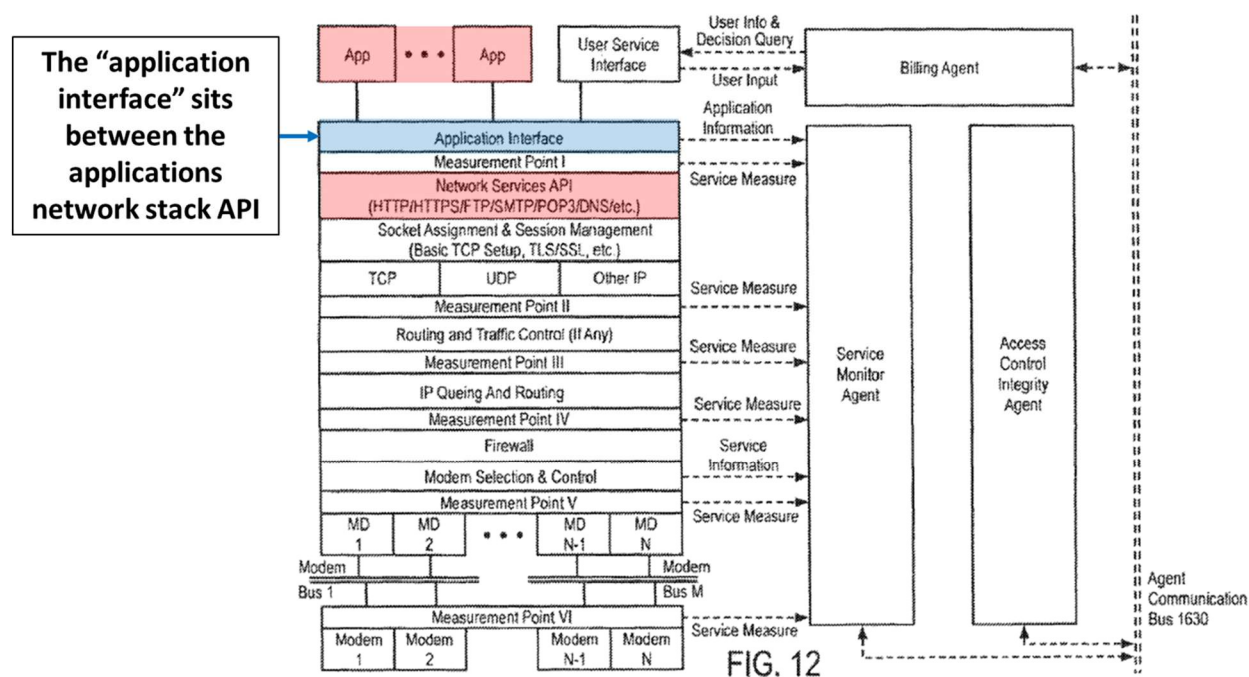


FIG. 28

EX1001, FIG. 28.

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; EX1003, ¶89.



EX1001, FIG. 12.

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; EX1003, ¶90.

These embodiments confirm the plain meaning of the claims: that “access” to APIs in the claims may be direct or indirect via an intermediary. EX1003, ¶¶91. 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

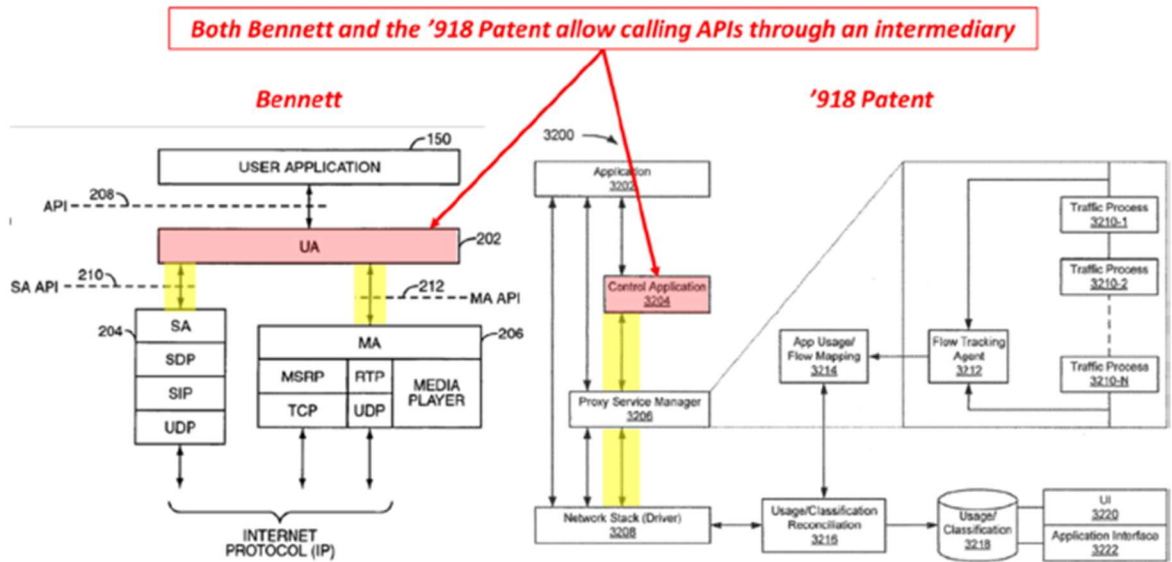
applications may indirectly access APIs including network stack APIs. EX1003, ¶91. For example, web applications running on a server do not typically directly access network stack APIs; rather, they interact with the network through “daemons” which are separate processes designed to open a socket using 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 “accessed” via the daemon by the web application. *Id.* Like the ’918 patent, the prior art similarly uses the term “accessible” broadly. EX1003, ¶91; *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. EX1003, ¶91.

Black-letter law supports Petitioner’s interpretation of the claims. For example, a narrower reading of “accessible” would impermissibly exclude embodiments from the proper scope of the claims. *CUPP Computing AS v. Trend Micro Inc.*, 53 F.4th 1376, 1381 (Fed. Cir. 2022) (“[A] claim interpretation that excludes a preferred embodiment from the scope of the claim is rarely, if ever, correct.”) (quoting *Accent Packaging, Inc. v. Leggett & Platt, Inc.*, 707 F.3d 1318,

1326 (Fed. Cir. 2013)). For an unmodified claim term such as “*accessible*,” it is also “improper to read limitations ... into the claims absent a clear indication in the intrinsic record that the patentee intended the claims to be so limited.” *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 913 (Fed. Cir. 2004).

Here, the claims do not recite the term “directly” nor do they otherwise preclude “indirect” access. EX1003, ¶92. 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. The Board should thus reject the argument now that accessibility must be “direct” as it does not “naturally align with the patent’s description.” *Phillips*, 415 F.3d at 1313 (citation omitted); *Renishaw PLC v. Marposs Societa’ per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998) (“The construction that stays true to the claim language” while “most naturally align[ing] with the patent’s description of the invention will be, in the end, the correct construction.”).

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. EX1003, ¶93. *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.

The Board agreed with Petitioner’s interpretation of the claims when instituting review in IPR2024-01396, holding that “Bennett teaches the claimed ‘first network stack Application Programming Interface (API), containing at least one first call accessible to each of a plurality of device applications.’” *Samsung*, IPR2024-01396, Paper 13 at 12-15. The Board should conclude the same here and find *Bennett* discloses the claimed “accessible” APIs.

e. 1[d]: “a second API ...”

“a data transfer request for a media object”

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; EX1003, ¶94.

“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”

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; EX1003, ¶95; *see* §VI.A.4.d. (1[c]) (describing a “*plurality of device applications*”).

The MA API 212 includes various “requests” to send and receive media, including “OPEN” requests (“*make a data transfer request for a media object*”). EX1041, ¶¶31, 33, 40, 45-46, 50, 54, FIGS. 6-7, Tables, 2-3; EX1003, ¶96.

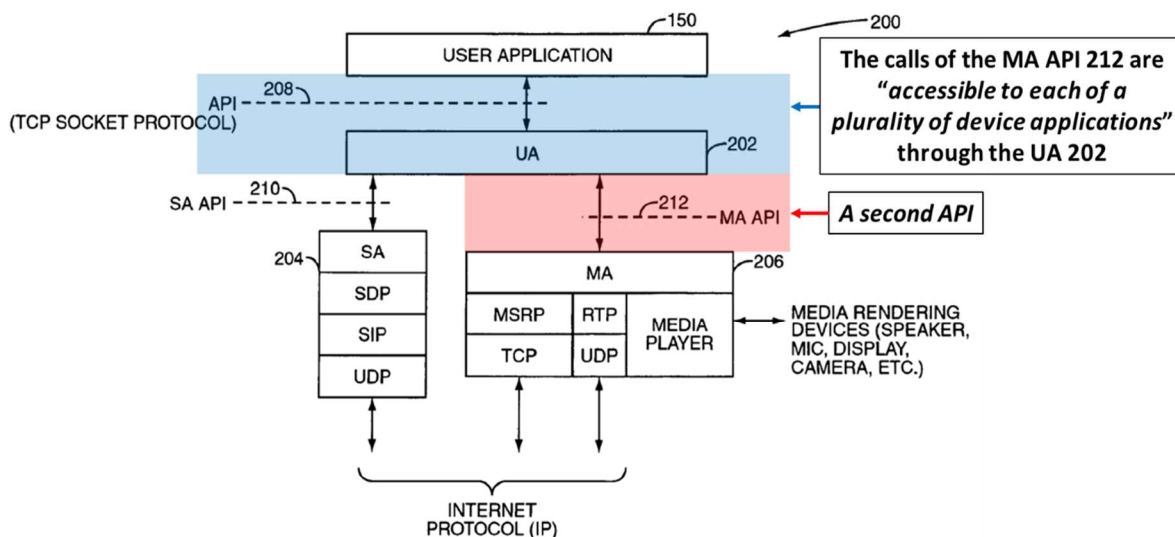
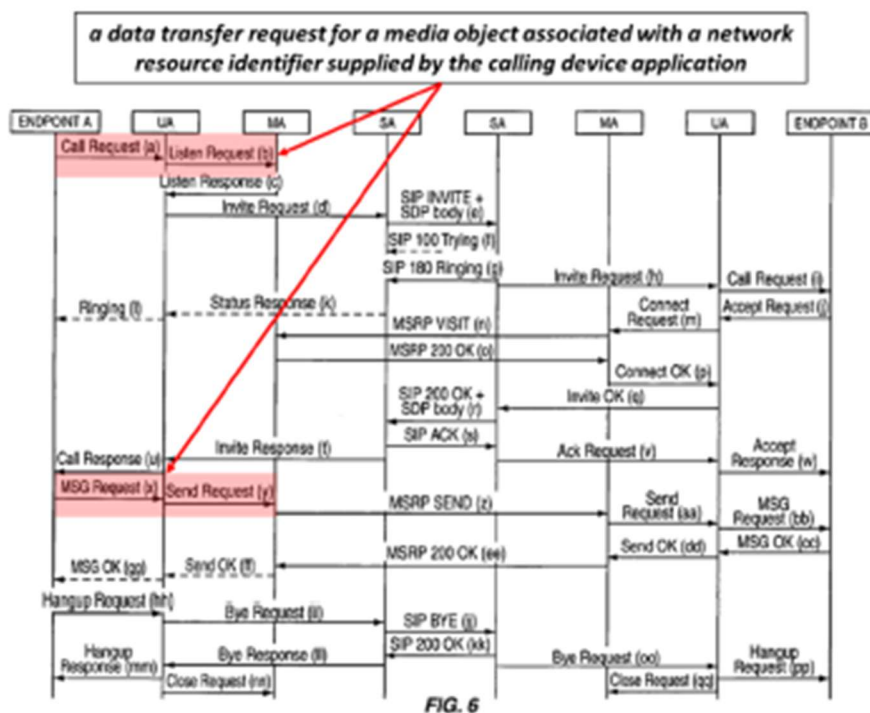
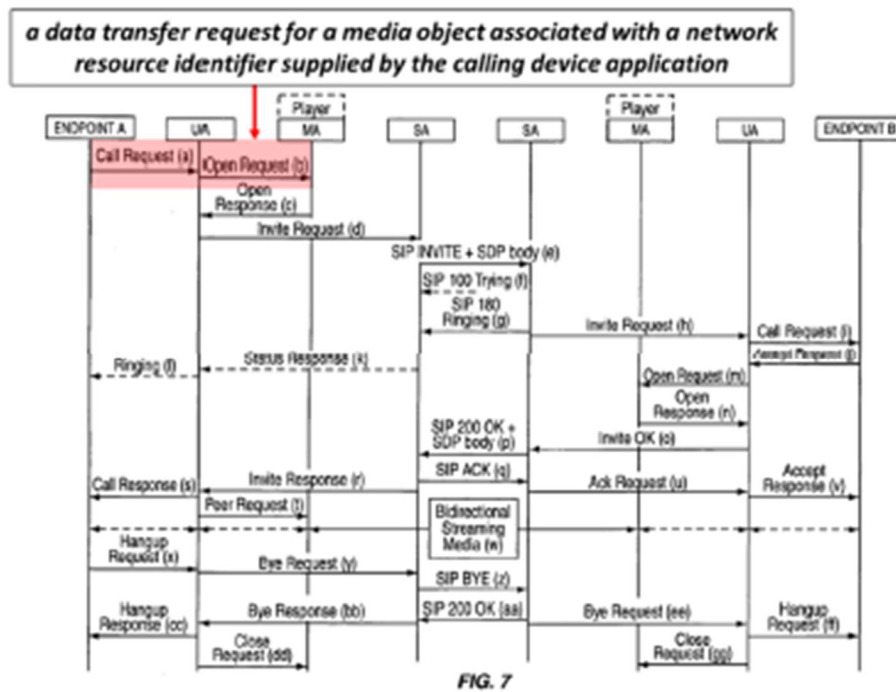


FIG. 3

EX1041, FIG. 3.



EX1041, FIG. 6.



EX1041, FIG. 7.

To the extent Patent Owner argues 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. EX1003, ¶97; *see* §VI.A.4.e. (1[d]). The process of applications indirectly calling APIs is instead disclosed in the ’918 patent in multiple embodiments. *See* §VI.A.4.e. (1[d]).

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

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; EX1003, ¶98.

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; EX1003, ¶99. Applications also provide a “userid” and a “port” to call via the CALL request (“*a network resource identifier*”). EX1041, ¶84, Table-1.

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; EX1003, ¶100.

“network resource identifiers” supplied by the calling application via the CALL request					
TABLE 1-continued					
		U/A API			
MESSAGE	USE	SYNTAX	PARAMETERS	PARAMETER DESCRIPTION	
Notify Response	Sent by IMS application to UA to acknowledge notify request	notify status...message	status...message	Indicates receipt of notify request, e.g. “OK”	
Publish Request	Sent by IMS application to IMS UA to publish change in the user's presence status	publish uri expire...time [autorefresh]	uri expiretime autorefresh	Address. Time before the publish expires in seconds. Optional flag instructing the UA to refresh the publish automatically when it expires. If the application does not want the UA to automatically refresh the publish, the flag is omitted.	
Publish Response	Sent by IMS UA to IMS application responsive to Publish request	publish uri expiretime status...message[status...code]	uri expiretime status...message status...code	Address. Sometimes, the server ignores the requested expire time and sets it to another value. This parameter returns the expiretime assigned by the server. Status of request indicating success (e.g. “OK”) or failure (e.g., “Failed”). Optional code indicating status of publish request, 200 if the request was successful or a failure code on failure.	
Call Request	Sent between IMS application and IMS UA to initiate MSRP and RTP sessions	call uriid [uriid[:resource[:port]]] call_type1...call_typeN	uriid host:port call...type	At the originating endpoint, the IMS application specifies a uriid to call when sending Call request if registered with a proxy. At the terminating endpoint the UA specifies the uriid of the calling party. At the originating endpoint, the UA specifies the host address and port to call, if not registered with a proxy. At the terminating endpoint the UA specifies the uriid of the network address and port designated by the calling party for the call. Type of call to be established, for example audio/mpeg or video/h263. Multiple call...types may be listed, e.g., audio/mpeg and video/h263 for video telephony	

EX1041, Table-1.

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

TABLE 3

MESSAGE	USE	SYNTAX	MA API	
			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 ...*”

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; EX1003, ¶101. 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 “OPEN” requests. EX1041, ¶¶31, 33, 40, 45-46, 50, 54, FIGS. 6-7, Tables 2-3 ; *see* §VI.A.4.e. (1[d]); EX1003, ¶101.

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* §VI.A.4.b. (1[a]); EX1003, ¶102.

The media object retrieved in *Bennett* is associated with the network resource identifier. EX1003, ¶103. As described above, requests from applications include a “*network resource identifier*” that is “*associated*” with the “*media object*.” *See* §VI.A.4.e. (1[d]); EX1003, ¶68. 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. EX1003, ¶103. For example, for OPEN requests, “media connections will be accepted only from the address specified in the OPEN request.” EX1041, ¶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.*

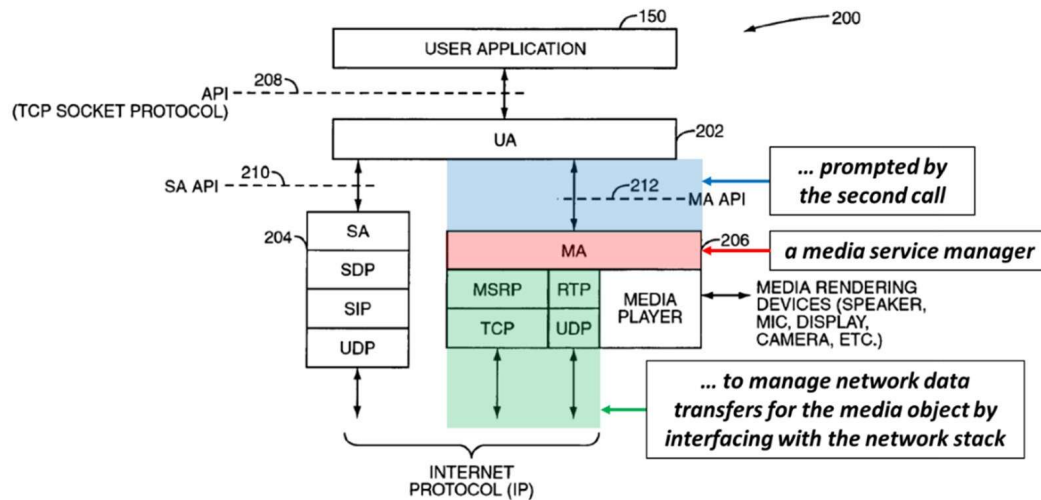
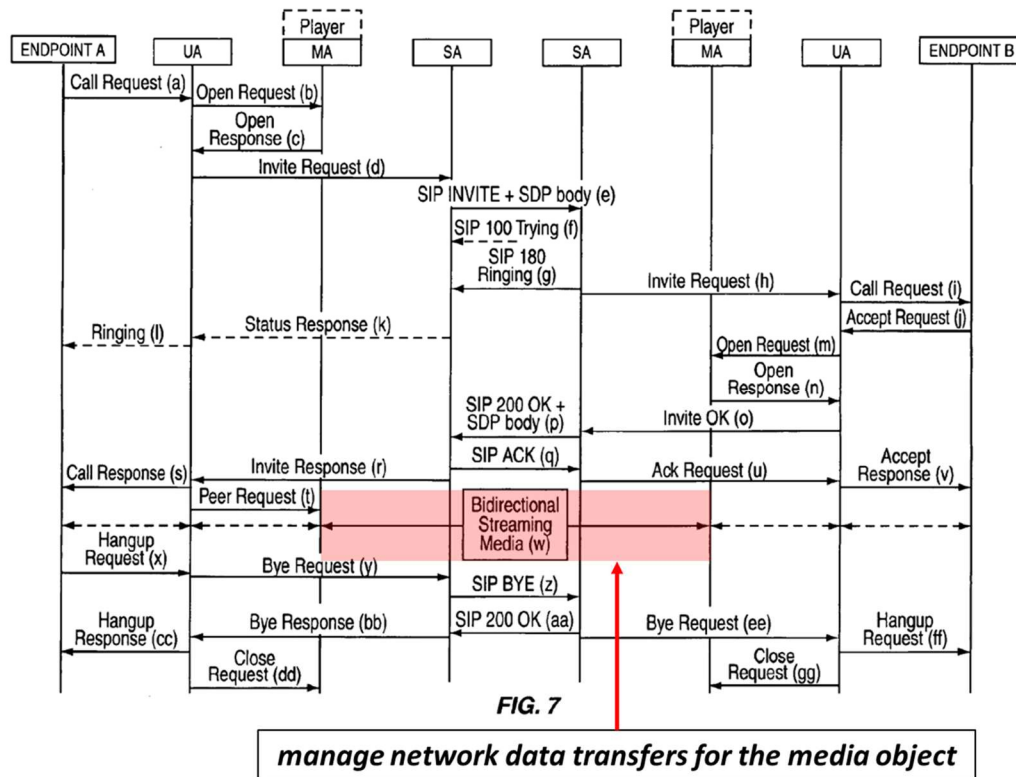


FIG. 3

EX1041, FIG. 3.



EX1041, FIG. 7.

g. 1[f]: “one or more service classification and measurement agents ...”

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; EX1003, ¶104. *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”

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; EX1003, ¶105. *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”***).

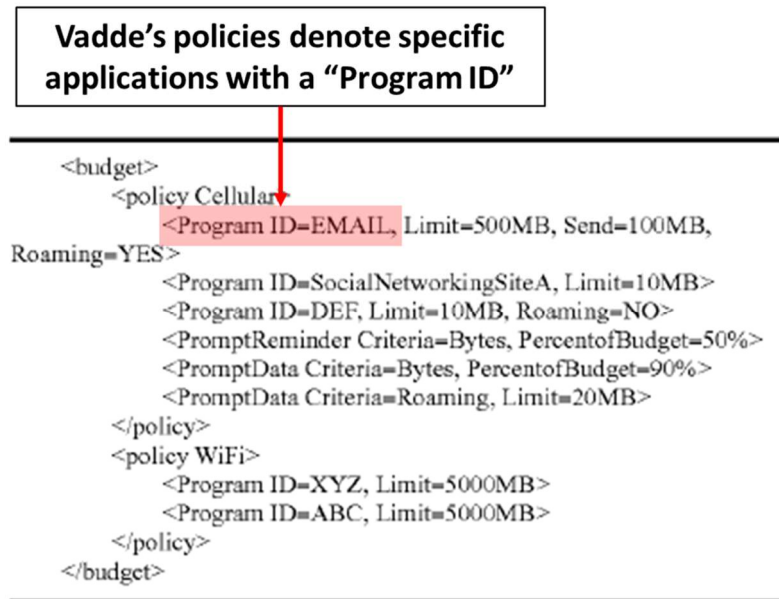
EX1042, ¶22. This includes monitoring transmitted data to “by receiving or accessing the attributes 112 or other statistics describing the data transmitted and/or received by each of the applications 110.” *Id.*, ¶22. Exemplary “attributes 112” include an “application name,” which is given a “Program 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.

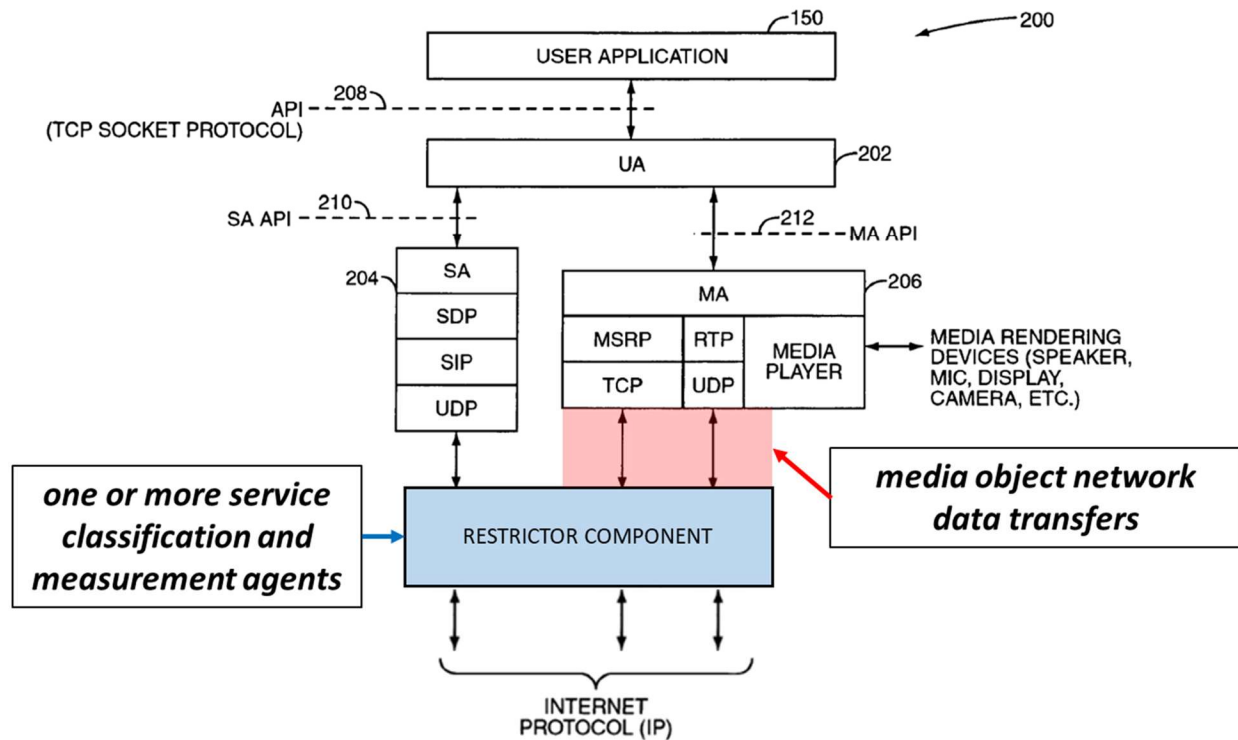
By monitoring data usage “corresponding” to each application, including accessing attributes that identify (e.g., classify) specific applications on a “per application” basis, and determining whether that usage exceeds a threshold (e.g., measure), restrictor component 122 discloses a “*service classification and measurement agent ... to associate wireless network data usage for ... the device application.*” EX1003, ¶106.

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 §VI.A.3; e.g., EX1042, ¶¶1, 34-35; EX1003, ¶107. 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 §VI.A.3; EX1003, ¶107;

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. EX1003, ¶107; 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”

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.

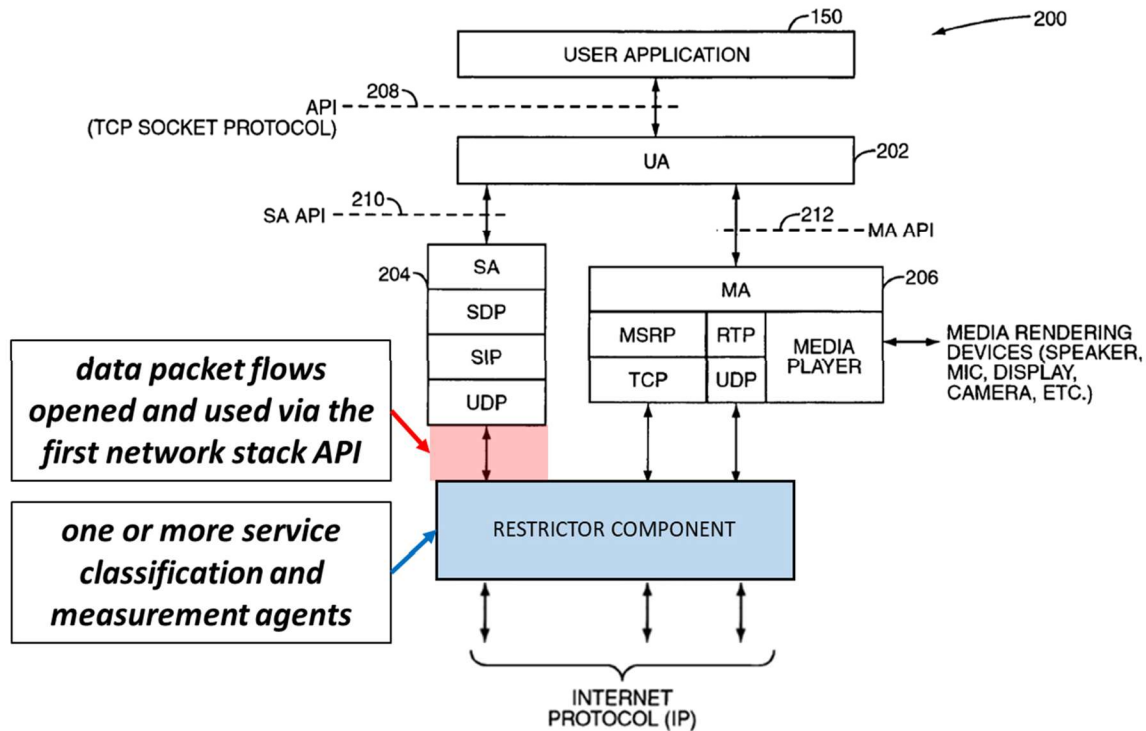
EX1042, ¶¶10, 15-16, 25-26; *see* §VI.A.3; EX1003, ¶108. As discussed, associating data usage on a per-application basis would identify data-intensive applications and those that draw excessive device power, as well as enable additional applications to be added while maintaining control of data usage. *See* §VI.A.3; EX1003, ¶108; EX1042 ¶¶9, 10, 16, 22, 23, 25, 29-30, 32.

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. EX1003, ¶109; EX1041, ¶17; *see* §VI.A.4.d (1[c]). *Rakoshitz* corroborates Dr. Houh’s testimony and describes that network traffic is “a ***flow of information*** or data or packets of information.” EX1046, 12:12-58, 15:57-67.

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; EX1003, ¶110.

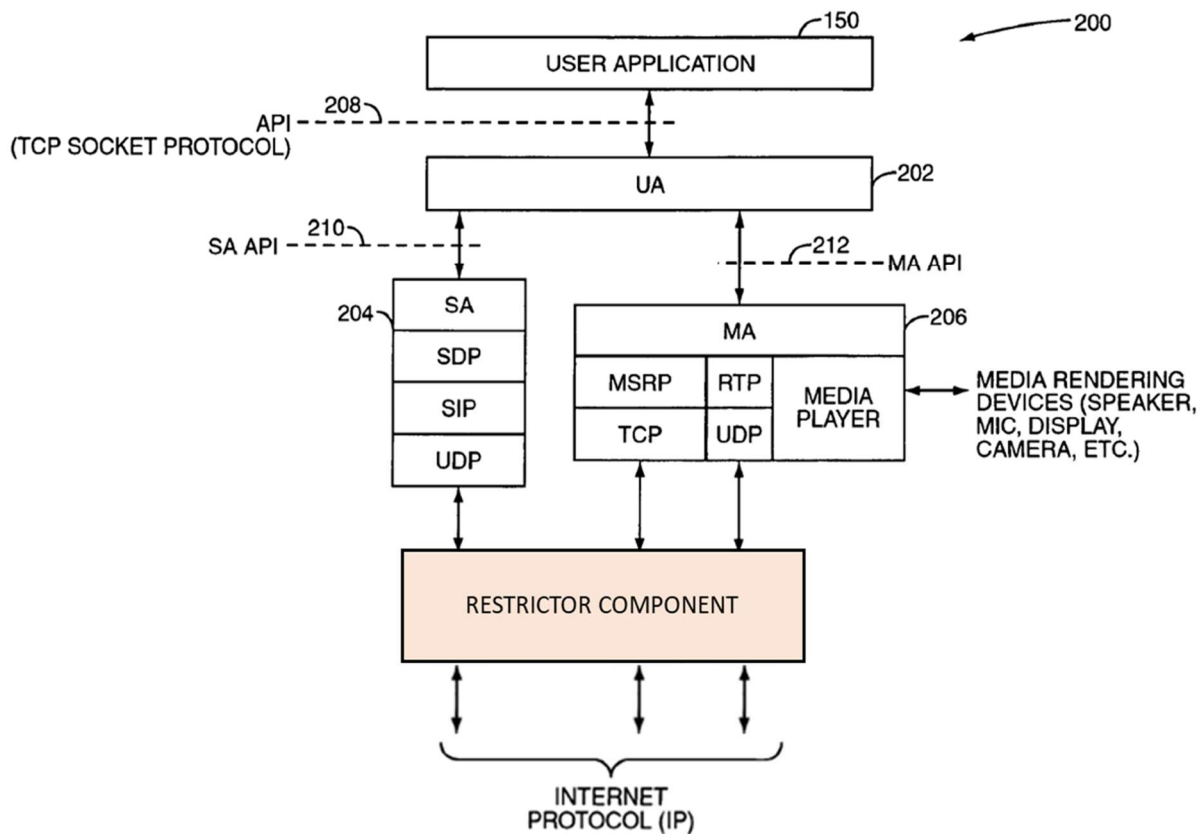


EX1041, FIG. 3 (as modified by *Vadde*).

“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”

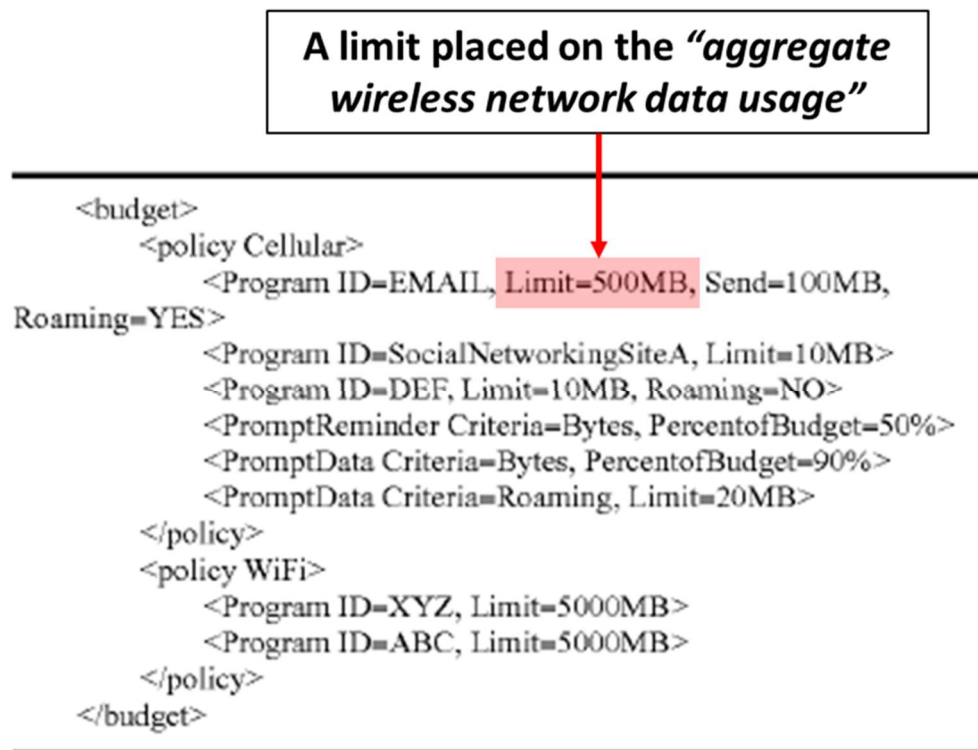
Vadde’s restrictor component 122 “monitors the data transmitted and/or received by the applications 110 and determines whether the data usage limits 116 corresponding to each of the applications 110 have been exceeded or are about to be exceeded.” EX1042, ¶¶22, 24-26, 29-32; EX1003, ¶111. In the combined 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. EX1003, ¶111; 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* §VI.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.” EX1003, ¶111.



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

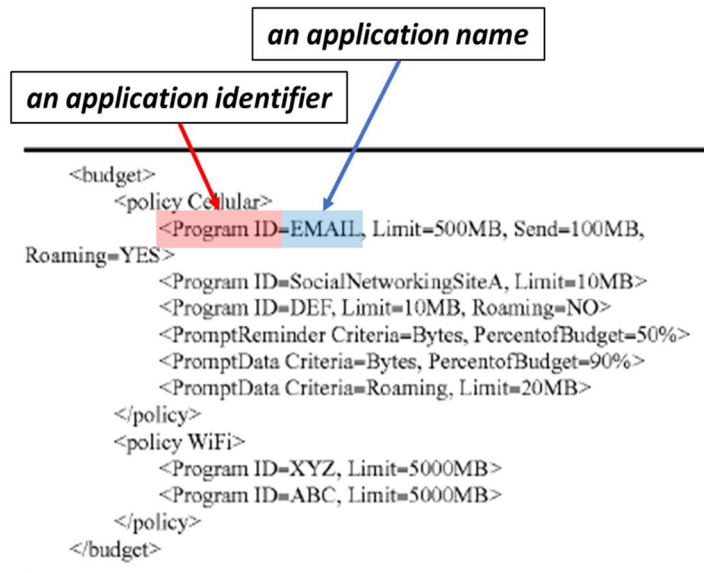
Vadde's "data usage limits 116" are set for "each" application and include "a maximum quantity of data to be transmitted and/or received by a particular application 110" ("*an aggregate wireless network data usage attributable to each of the plurality of device applications*"). EX1042, ¶¶10, 16, 18, 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 §VI.A.3; EX1003, ¶112.



EX1042, ¶32.

5. Claim 2

Vadde discloses monitoring data traffic and “defin[ing] ... one or more attributes” in the data. EX1042, ¶24. 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 §VI.A.4.g (1[f]); EX1003, ¶113. 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.*” EX1042, ¶¶24, 32; see §VI.A.4.g (1[f]); EX1003, ¶113. *Bennett-Vadde* therefore renders claim 2 obvious. EX1003, ¶113.



EX1042, ¶32.

6. Claim 3

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* §VI.A.4.e (1[d]); EX1003, ¶114. 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.

"a proxy to the source of the data to be transferred"				
Call Request	Sent between IMS application and IMS UA to initiate MSRP and RTP sessions	call userid [userid@remotehost[:port]] call_type1...call_typeN	userid	At the originating endpoint, the IMS application specifies a userid to call when sending Call request if registered with a proxy. At the terminating endpoint the UA specifies the userid of the calling party.
			host:port	At the originating endpoint, the UA specifies the host address and port to call, if not registered with a proxy. At the terminating endpoint the UA specifies the userid of the network address and port designated by the calling party for the call.
			call_type	Type of call to be established, for example audio/amr or video/h263. Multiple call_types may be listed, e.g., audio/amr and video/h263 for video telephony

EX1041, Table-1.

Bennett's "*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; EX1003, ¶115; *see* §VI.A.4.e (1[d]).

URLs would indicate “*the media object to be transferred, in particular*” because the file name of the media object would typically comprise part of the URL. EX1003, ¶115. Indeed, *Riggs* discloses that “the global address of content” 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.

A POSITA would have understood and found obvious that URLs indicate “*media objects*”

One way to identify a publisher associated with content being played back is use of an “authority” section of a feed URL (e.g., a pointer received via media feed 159) that references the media content. For example, if the user is playing back an episode from NBC’s *Heroes* television show, then the feed URL for the *Heroes* feed may be something like “http://www.nbc.com/heroes/feed.xml”. In this case, the authority (e.g., publisher id information) is “www.nbc.com”, which is the publisher identifier.

EX1043, 5:54-62.

7. Claim 8

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 MA 206 to provide the MA 206 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; EX1003, ¶116. 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*”). EX1041, ¶¶55, 73, 79, FIG. 7; EX1003, ¶116.

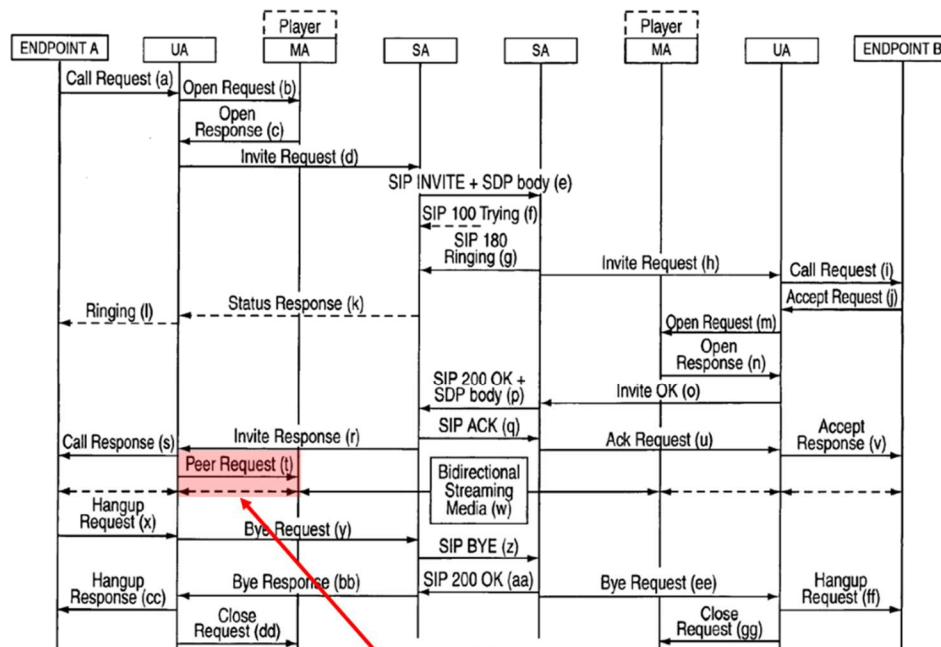


FIG. 7

map the data transfer request and network resource identifier to one or more data flow connections

EX1041, FIG. 7.

8. Claim 9

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. *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; EX1003, ¶117. This may occur during, for example, a videoconferencing session following a CALL request. See §VI.A.4.f (1[e]); EX1041, ¶¶17, 18, 24. Bennett’s FIGS. 3 and 9 depict examples of this process.

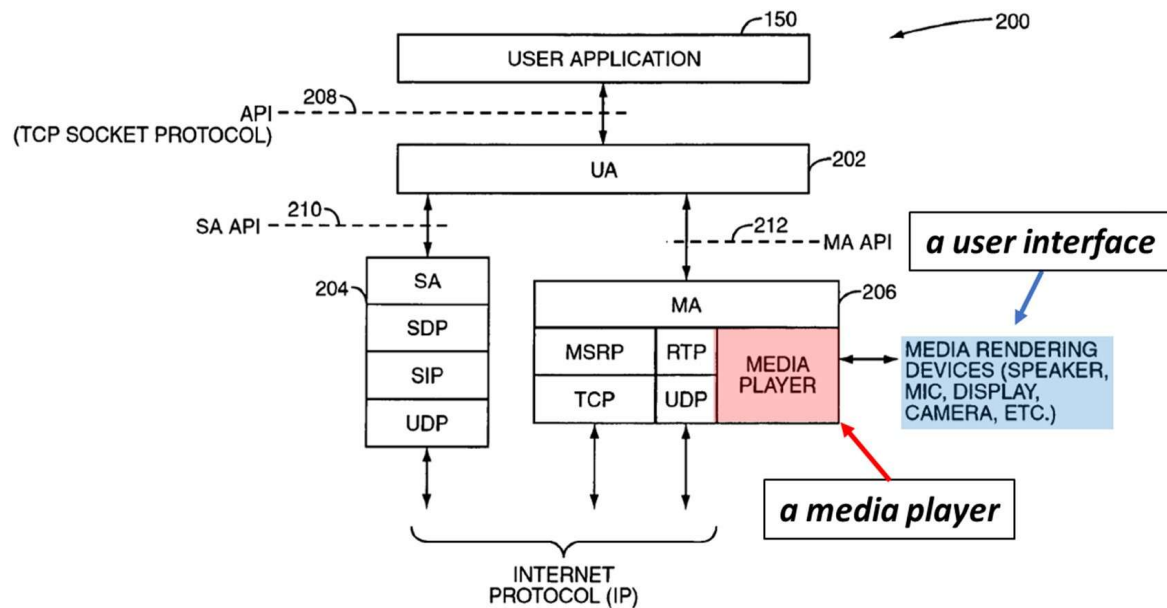
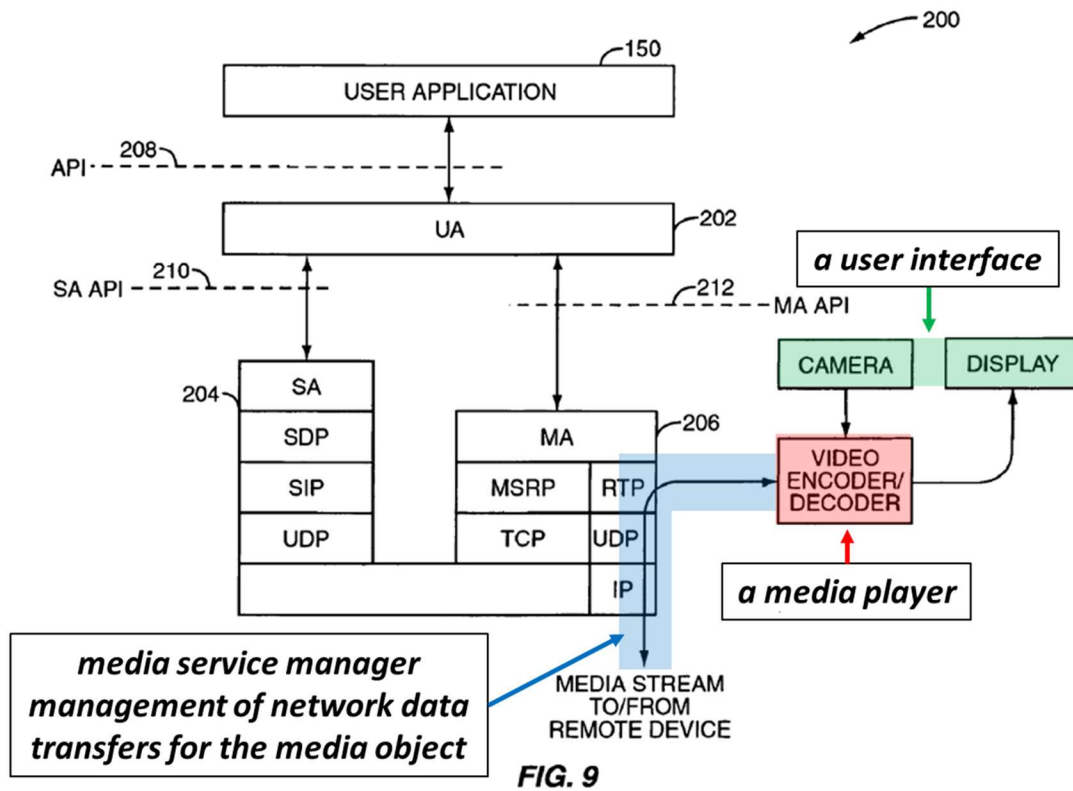


FIG. 3

EX1041, FIG. 3.



EX1041, FIG. 9.

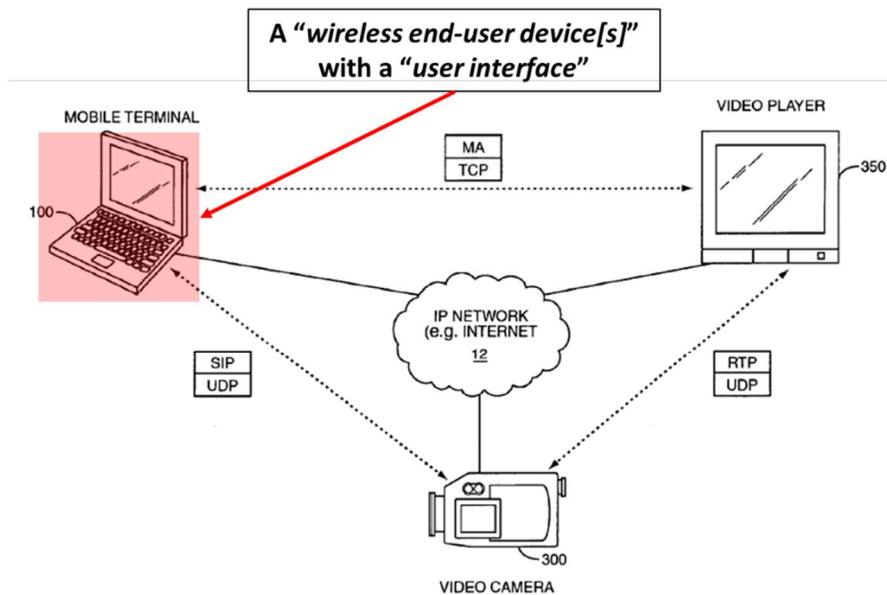
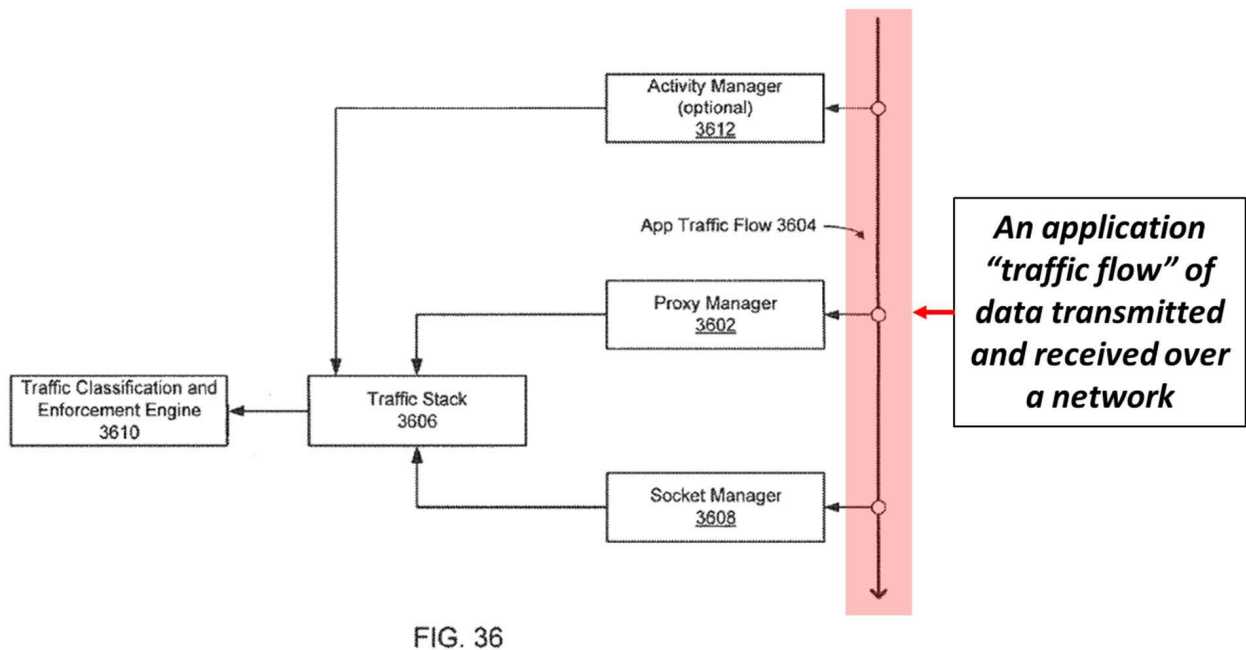


FIG. 11

EX1041, FIG. 11.

9. Claim 13

The '918 patent describes that applications generate “*traffic flows*” (e.g., data transmitted/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; EX1003, ¶118.



EX1001, FIG. 36.

As described above, *Vadde*’s “policy” 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 §VI.A.4.g (1[f]). *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* §VI.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). EX1003, ¶119.

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* §VI.A.4.e (1[d]); EX1003, ¶120.

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* §VI.A.4.g (1[f]); EX1003, ¶121. 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; EX1003, ¶121.

10. Claims 14 and 19

The following features are disclosed and/or rendered obvious for the reasons as discussed in the analysis for the corresponding claim listed in the table below.

EX1003, ¶¶122-124.

Claim 14 is substantively identical to claim 1 but recites a method that implements claim 1's features, with the addition of "operating" the APIs and media service manager. EX1003, ¶123.

Claim 19 is device that is likewise substantively identical to claim 1 but more broadly recites the APIs "allow[ing]" the claim 1's feature, while also incorporating the limitations of claims 3, 8, and 2. Claim 19's "service classification agent" is taught and rendered obvious for the reasons explained in claim 1 with respect to "service classification and measurement agents." EX1003, ¶125.

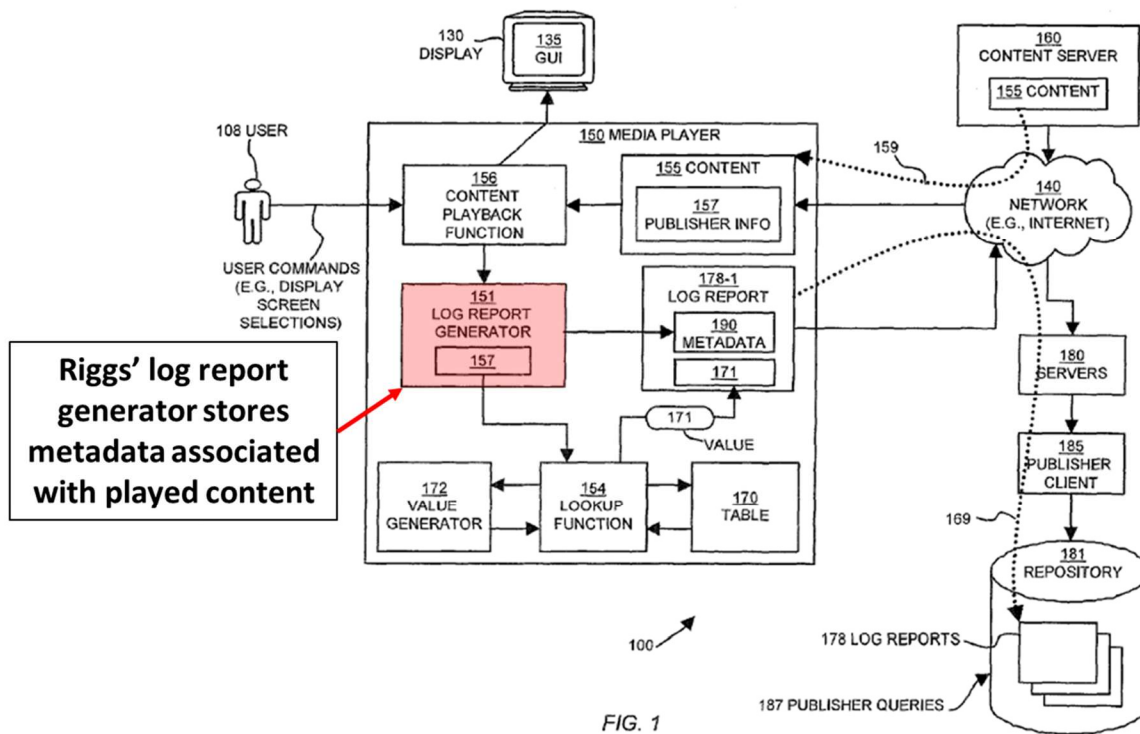
Claim	Corresponding Claim
14[pre]	1[pre], 1[a]
14[a]	1[b], 1[c]
14[b]	1[d]
14[c]	1[e]
14[d]	1[f]
19[pre]	1[pre]

19[a]	1[a]
19[b]	1[b]
19[c]	1[c]
19[d]	1[d], 3
19[e]	1[e], 8
19[f]	1[f], 2

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

1. Overview of *Riggs*

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; EX1003, ¶125.



EX1043, FIG. 1.

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

It would have been obvious 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; EX1003, ¶126.

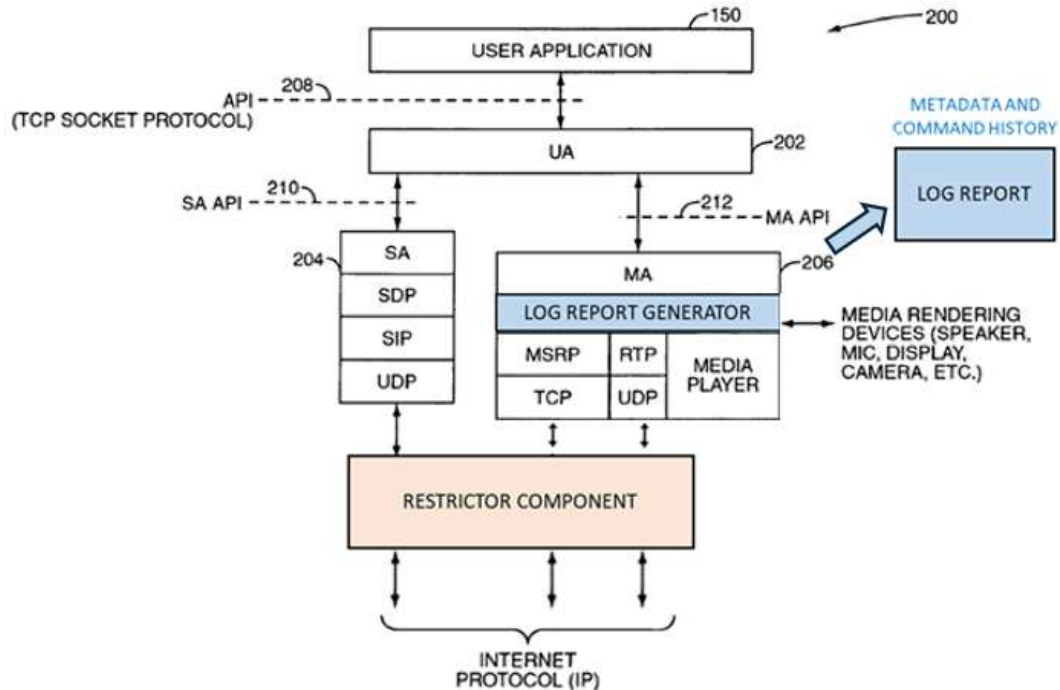
Specifically, *Riggs* discloses that, by “record[ing] events such as occurrence of playback commands and related playback information,” a publisher can “identify corresponding portions of such content that are most appealing to a viewer.” EX1043, 1:50-57, 14:27-36. Incorporating *Riggs*'s techniques 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; EX1003, ¶127. 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.

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; EX1003, ¶128.

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 within *Bennett*’s MA 206. EX1003, ¶129. 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. *Riggs*’s log generator is complementary to *Vadde*’s 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 implement *Riggs* this way because the log generator and restrictor component perform related functions (e.g., tracking application usage and reporting on the same). EX1003, ¶129. 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. EX1003, ¶129. *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. EX1003, ¶129.



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

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). *KSR*, 550 U.S. at 417. 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). EX1003, ¶130. Indeed, the modification would have involved routine programming ability that would have been well within the skill of a POSITA. *Id.*

3. Claim 4

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; EX1003, ¶131.

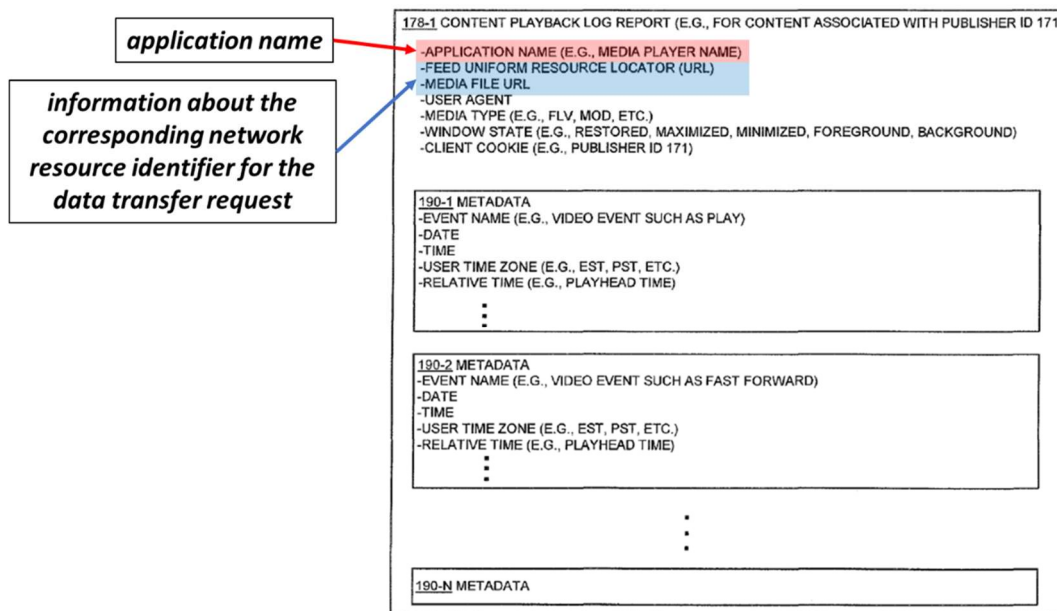
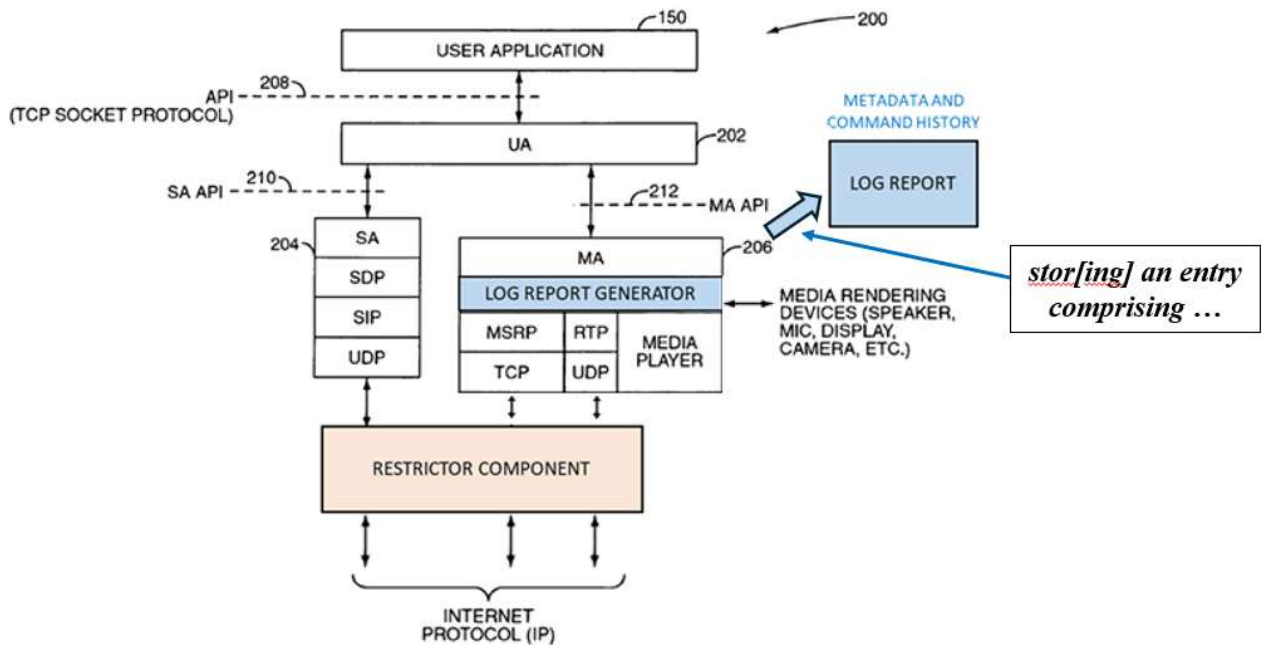


FIG. 2

EX1043, FIG. 2.

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 §VI.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*. EX1003, ¶132. 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 §VI.B.2. Doing so would have provided the benefits described above, including gaining increased insight into a user’s data usage patterns. See §VI.B.2; EX1003, ¶132.



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

4. Claim 5

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 §VI.B.3 (4) (describing “log reports”); EX1003, ¶133.

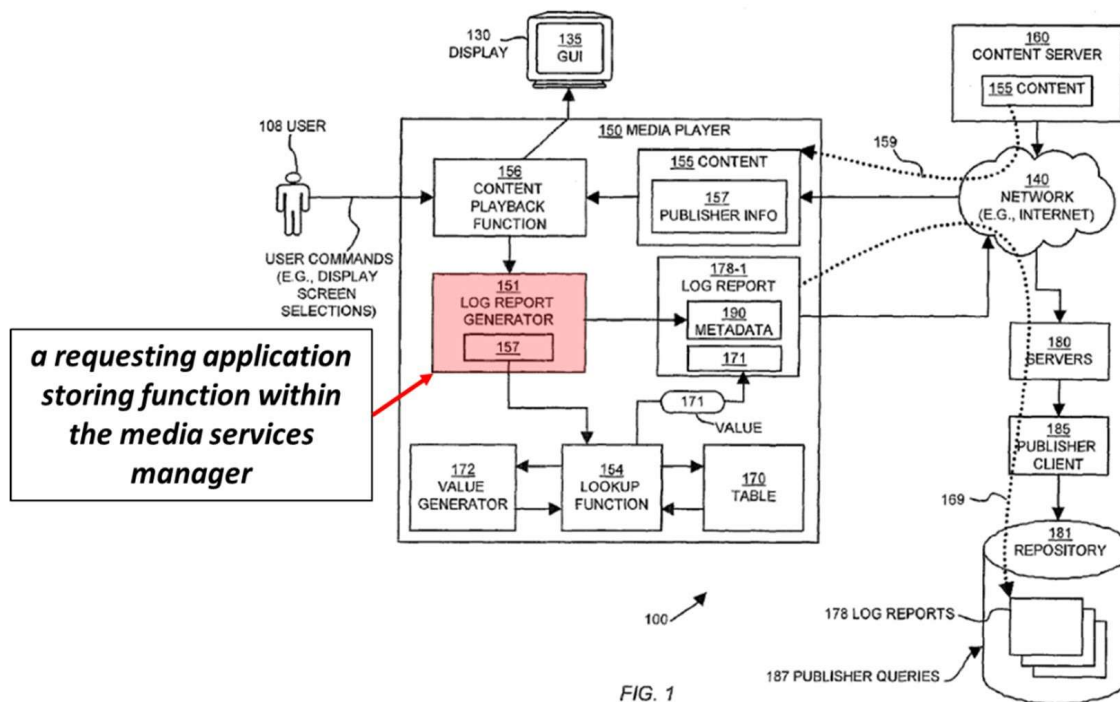
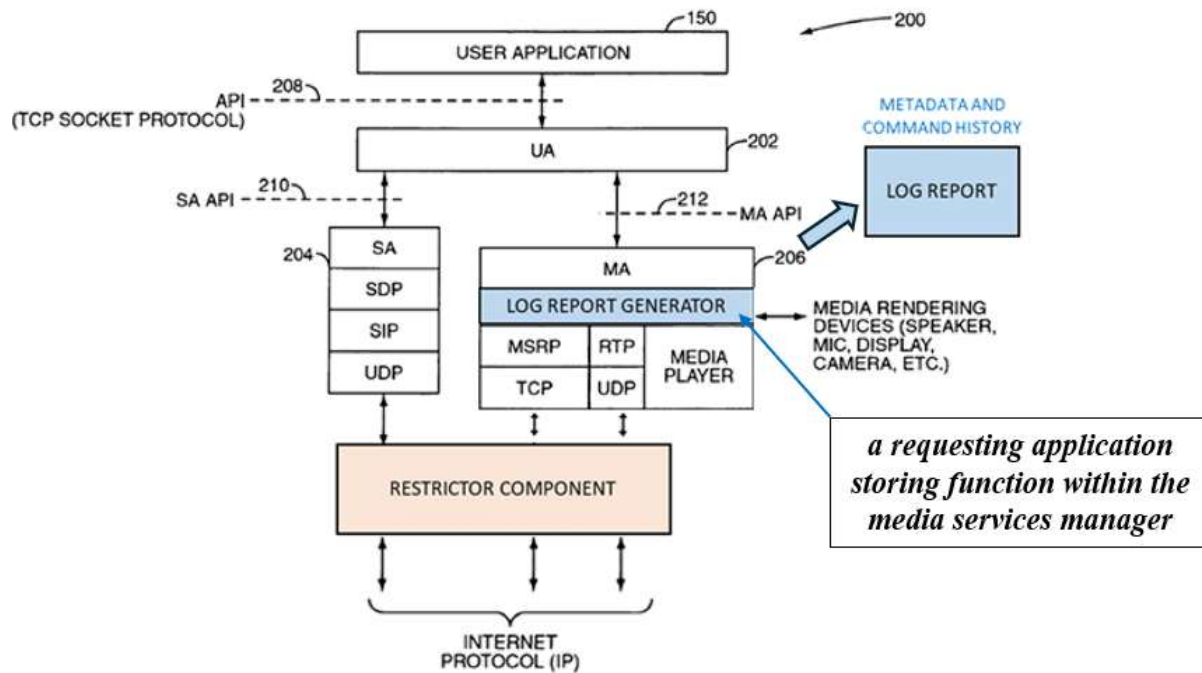


FIG. 1

EX1043, FIG. 1.

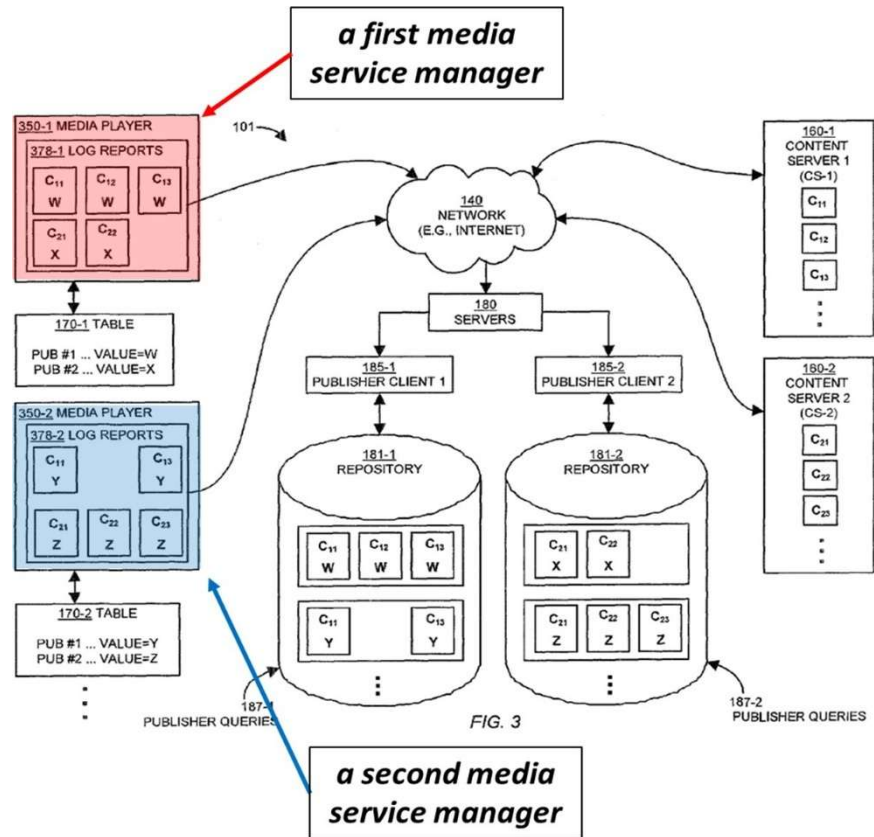


EX1041, FIG. 3 (as modified by *Vadde and Riggs*).

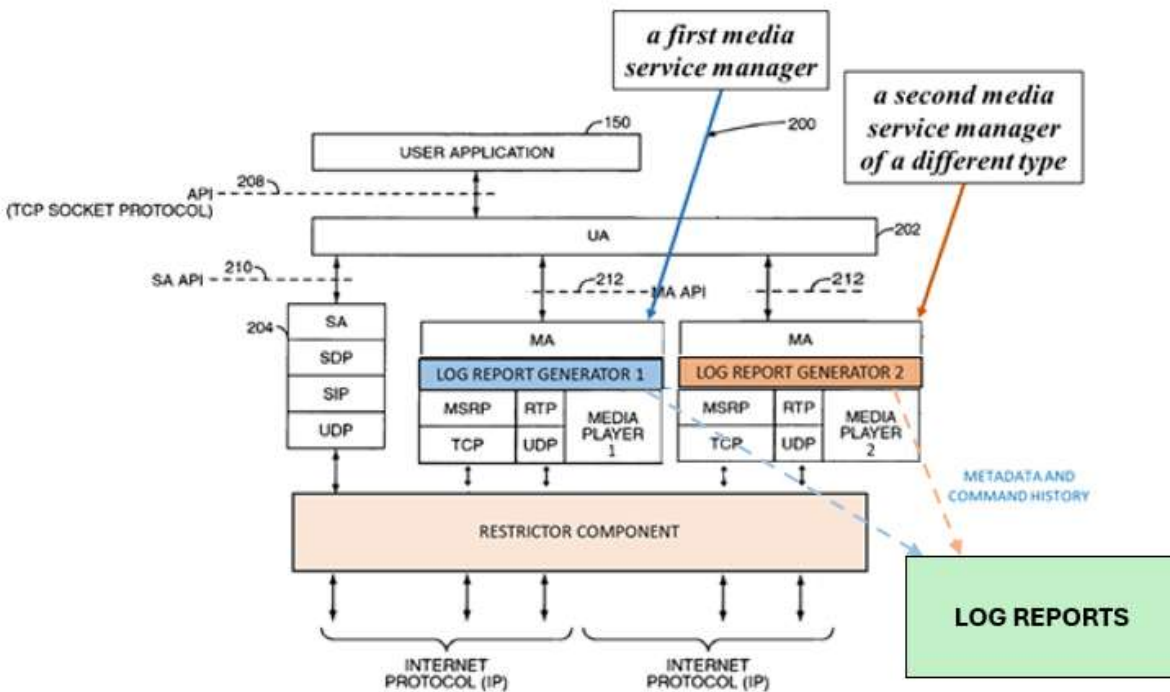
5. Claim 6

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; EX1003, ¶134. 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,
12:43-57; EX1003, ¶134.



EX1043, FIG. 3.



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

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 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; EX1003, ¶135; *see In re Harza*, 274 F.2d 669, 671 (CCPA 1960)

(holding that mere duplication of parts absent some unexpected result carries no patentable significance). 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.

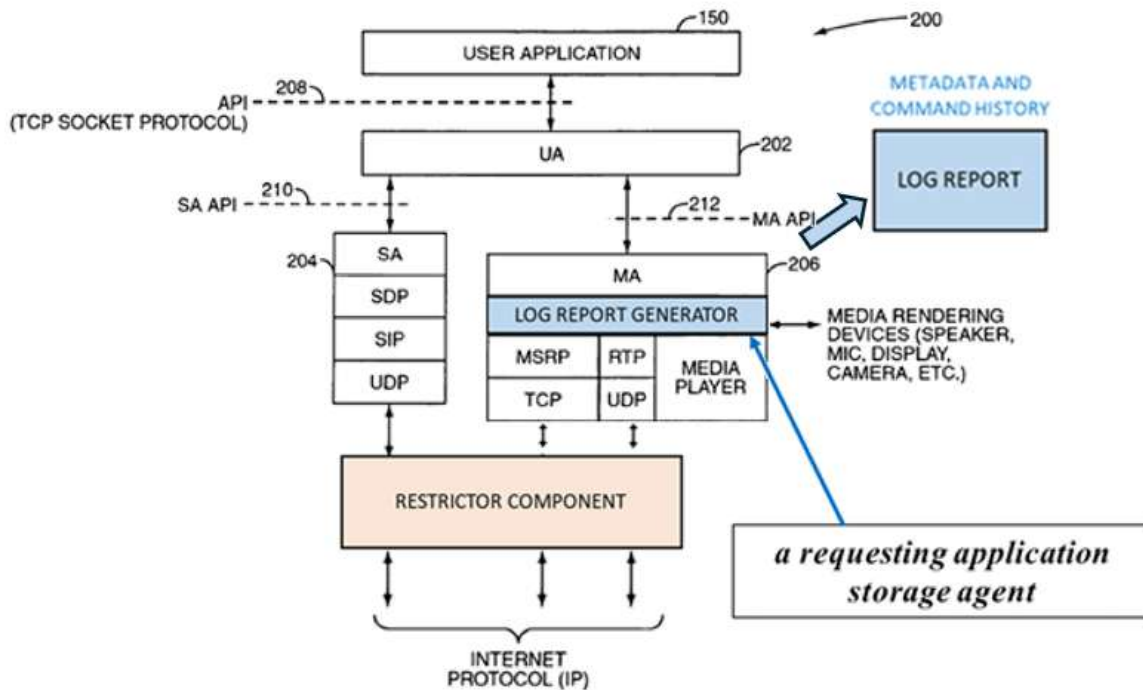
6. Claim 11

a. 11[a]

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 §§VI.B.3 (4) & VI.B.4 (5). A POSITA would have understood that a “function” and “agent” are synonymous terms for software components that perform operations in computer code. EX1003, ¶136. 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.

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* §VI.A.4.g (1[f]); EX1003, ¶137.



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

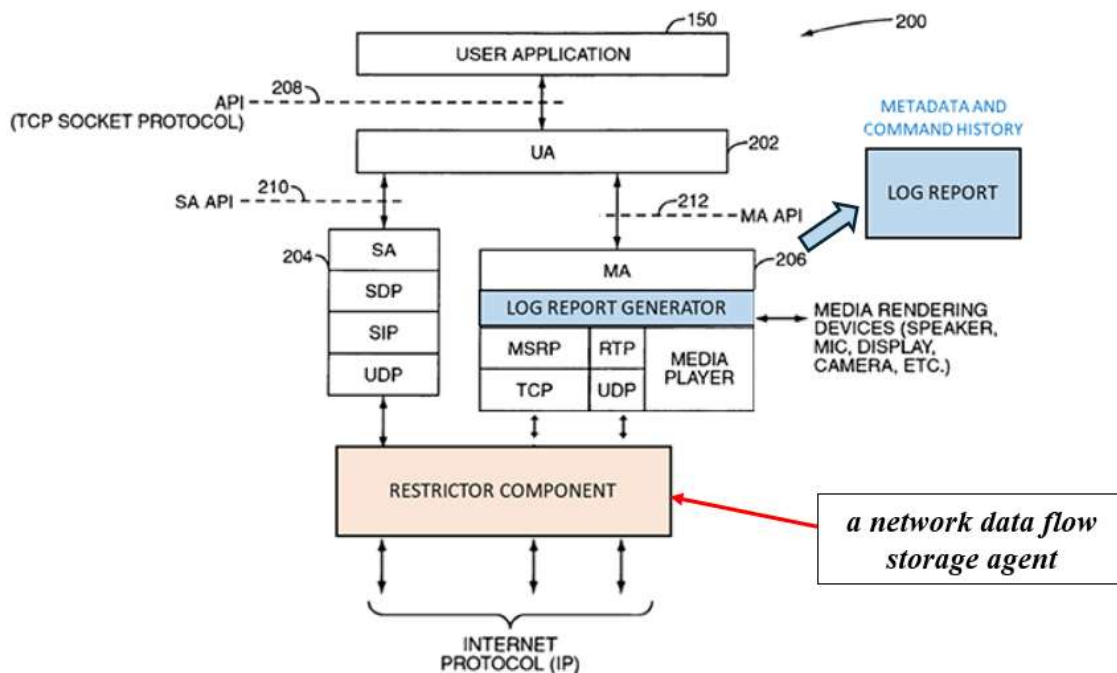
b. 11[b]

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; EX1003, ¶138.

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.

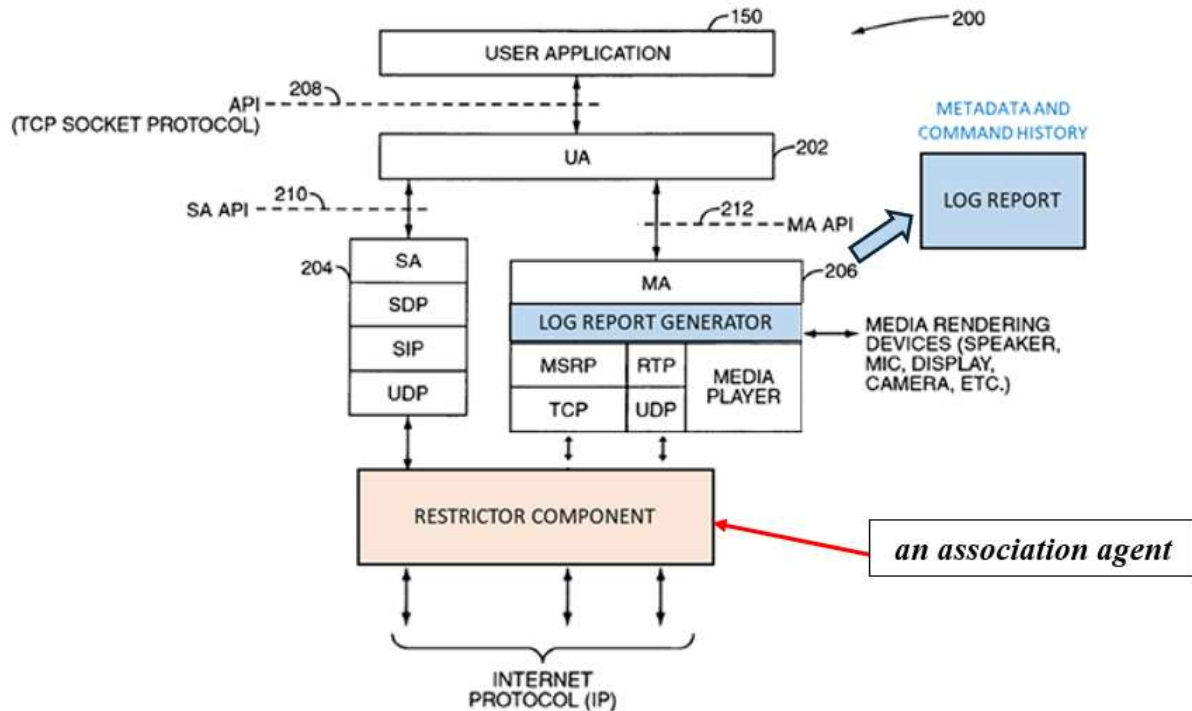
Additionally, *Bennett*’s requests include “a network address of a remote host from which media connections will be accepted,” for example, a “fully qualified network address” (e.g., an IP address—“*network data flow identification information*”). EX1041, ¶¶34, 50-56, Table-3; *see* §VI.A.4.e (1[d]); EX1003, ¶139. 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* §VI.A.4.g (1[f]); EX1003, ¶139.



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

c. 11[c]

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* §VI.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; EX1003, ¶140. 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*).

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*.” EX1003, ¶141. 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. EX1003, ¶141. *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. EX1003, ¶141. The modification would have involved nothing more than routine programming skill with a reasonable expectation of success. EX1003, ¶141.

7. Claims 15-17

The following claims are rendered obvious for similar reasons as discussed in the analysis for the corresponding claim listed in the table below. EX1003, ¶¶142-144.

Claim 15 is a device claim that is substantively identical to claim 1 but more broadly recites the APIs “allow[ing]” claim 1’s feature, while also incorporating the limitations of claims 2-4. Claim 15’s “service classification agent” is taught

and rendered obvious for the reasons explained in claim 1 with respect to the “service classification and measurement agents.” EX1003, ¶143.

Claims 16 and 17 are substantively identical to claims 5 and 6, respectively. EX1003, ¶144.

Claims	Corresponding Claim
15[pre]	1[pre]
15[a]	1[a]
15[b]	1[b]
15[c]	1[c]
15[d]	1[d], 3
15[e]	1[e]
15[f]	1[f], 2, 4
16	5
17	6

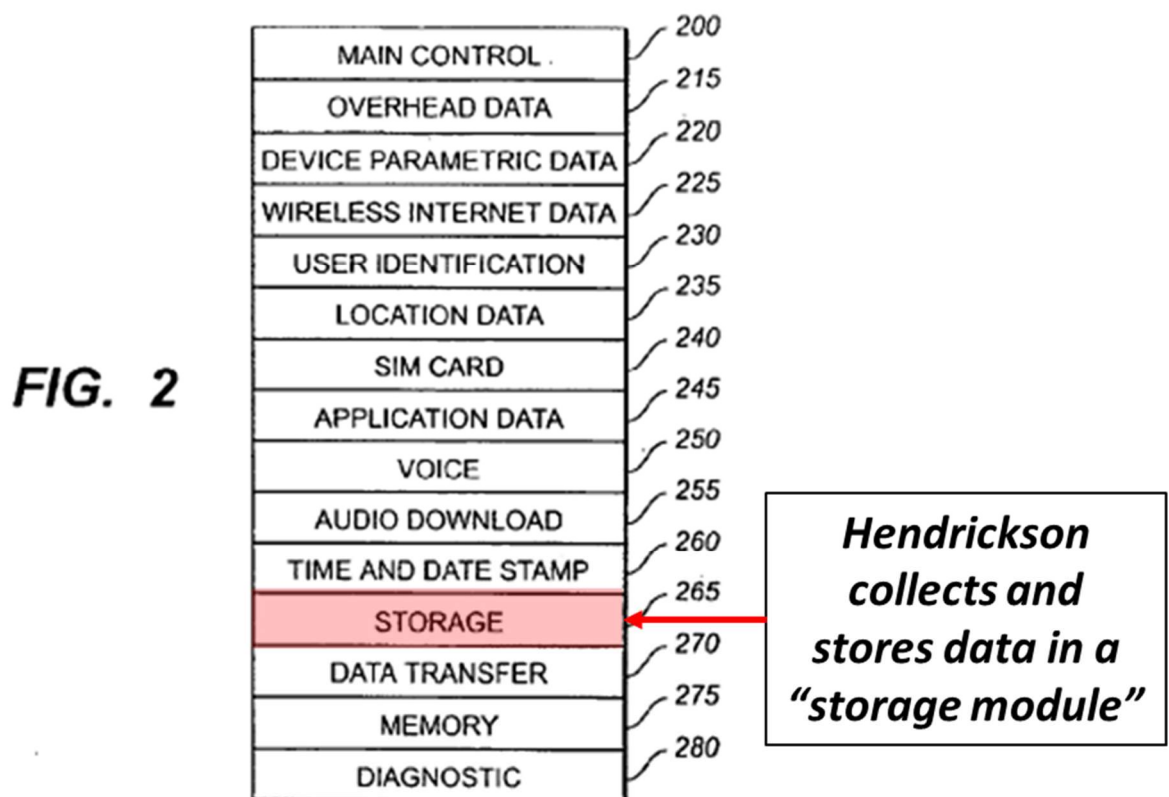
C. Ground 3: Claims 7, 12, and 18 Are Obvious over *Bennett* in View of *Vadde*, *Riggs*, and *Hendrickson*

1. Overview of Hendrickson

Hendrickson discloses a system for measuring “wireless device and wireless network usage and performance metrics” that 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.

Data is stored 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; EX1003, ¶145.



EX1054, FIG. 2.

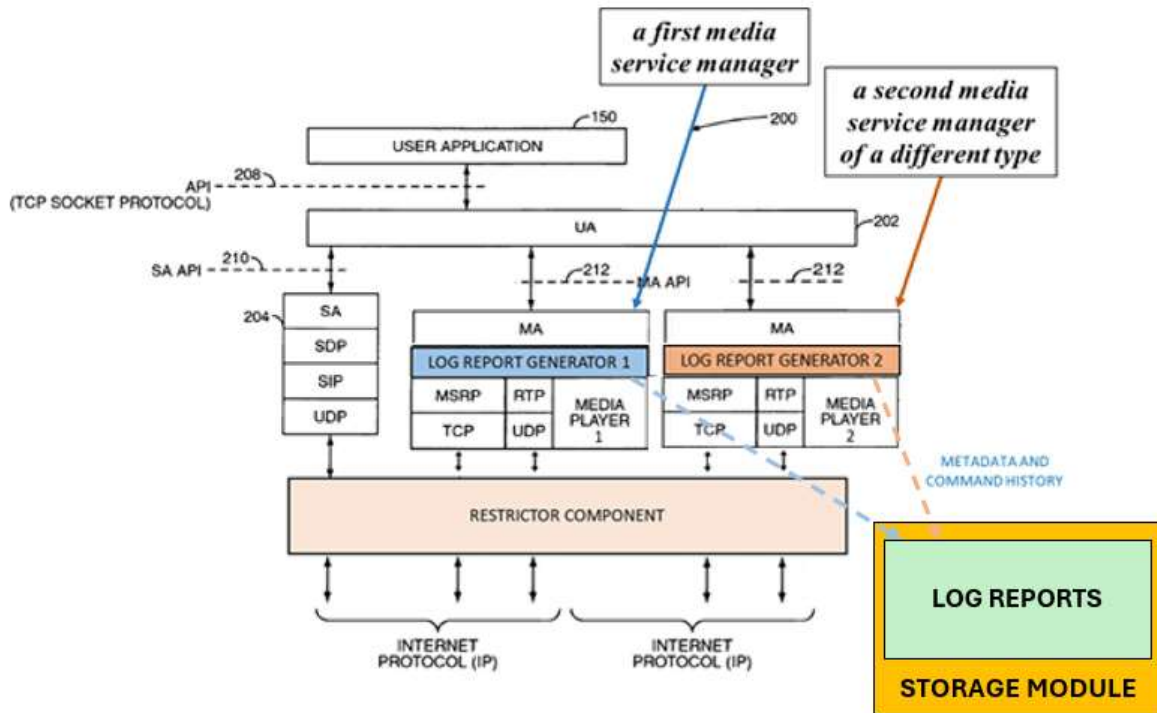
2. Motivation to Combine *Bennett-Vadde-Riggs* and *Hendrickson* with Reasonable Expectation of Success

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* §§VI.B.3 (4), VI.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* §§VI.B.3 (4), VI.B.5 (6); EX1003, ¶146. 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) and would have recognized the benefits of doing so, including facilitating synchronization of data with remote storage and allowing users to access and search the records storage in the storage module 265. EX1003, ¶146.

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). *KSR*, 550 U.S.

at 417. 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). EX1003, ¶147. 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”); EX1003, ¶147.



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

3. Claim 7

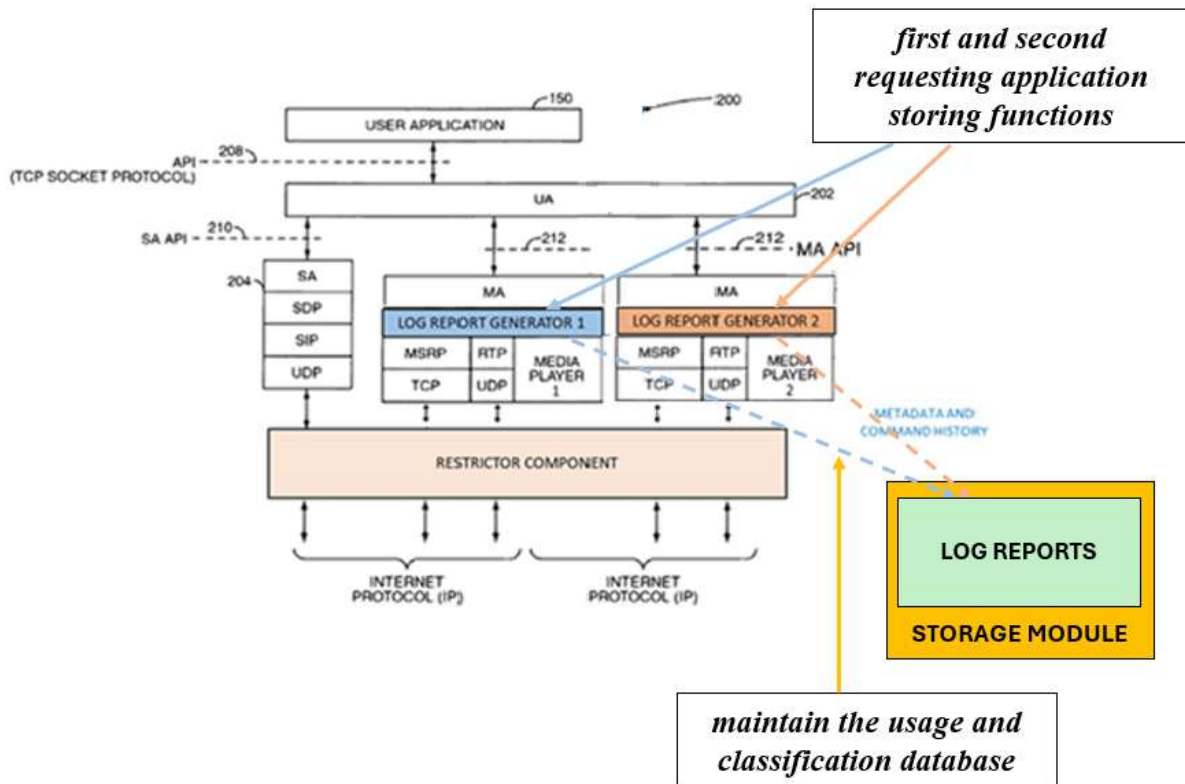
The '918 patent does not define a “*usage and classification database*” but rather describes that “the results of the service usage classification or accounting can be stored in a local device (or operating system) database” and that the stored information “can be provided to other applications, operating system service functions, other device software functions, or network-based service classification or accounting functions.” EX1001, 113:46-55; EX1003, ¶148. 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* §§VI.B.3 (4), VI.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* §VI.B.4 (5); EX1003, ¶98.

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; EX1003, ¶149. Moreover, the *Bennett-Vadde-Riggs* device would have included a means for electronic storage of files, like log reports. EX1041, ¶29; EX1042, ¶¶11-14, 17; EX1043, 14:37-15:2, FIG. 4; EX1003, ¶149. *Riggs*’s log reports therefore renders obvious the “***usage and classification database.***” EX1003, ¶149.

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; EX1003, ¶150. 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* §§VI.B.3 (4), VI.B.5 (6), II.C; EX1003, ¶150.



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

4. Claim 12

a. 12[a]

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 §VI.B.2; EX1003, ¶151. In the combination with *Hendrickson*, *Riggs*'s log report generators 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* §§ VI.A.4.f (1[e]) (describing *Bennett-Vadde* disclosure of a media service manager managing network data transfers for the media object), VI.C.2 (describing implementation as a “database”), VI.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; EX1003, ¶151.

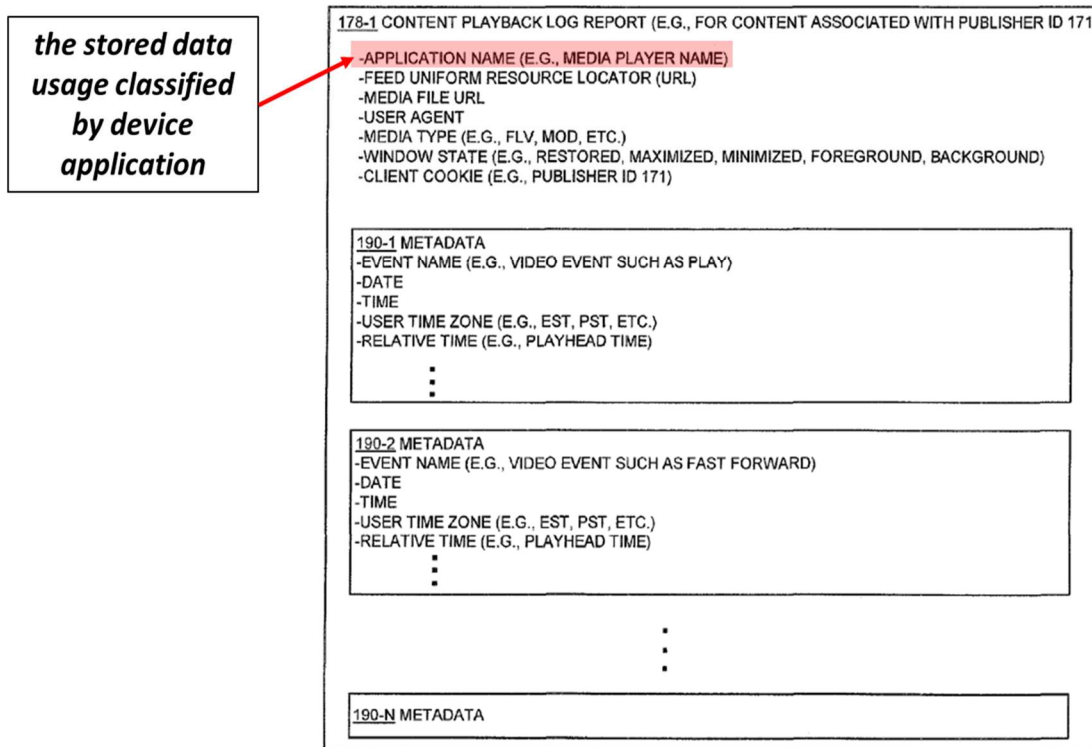


FIG. 2

EX1043, FIG. 2.

b. 12[b]

As discussed with claim 9 above, *Bennett* discloses “*a user interface.*”

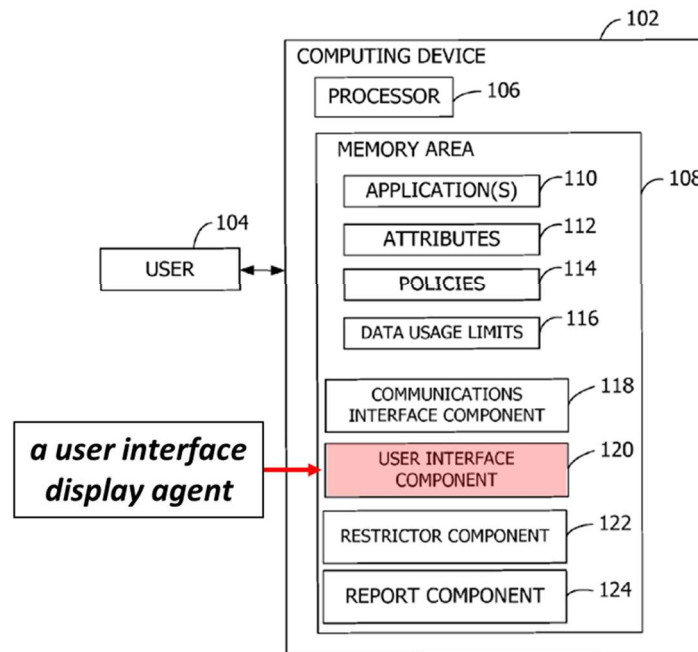
EX1041, ¶¶2, 25, 76, FIGS. 3, 9, 11; *see* §VI.A.8 (9); EX1003, ¶152.

c. 12[c]

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; EX1003, ¶153.

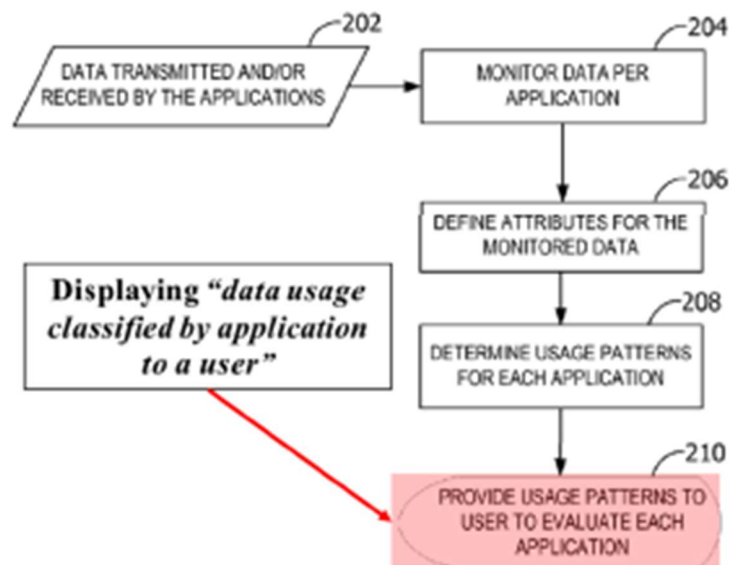
In the *Bennett-Vadde-Riggs-Hendrickson* combination, generated log reports for each application—including metadata and data usage— would have been displayed to the user via the “user interface component 120” and a “user interface” (“*the data usage classified by application to a user*”). EX1003, ¶154. 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

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 §VI.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; EX1003, ¶155.

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 §VI.C.3 (7); EX1003, ¶156.

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 §§VI.A.4.g (1[f]), VI.A.5 (2), VI.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 §§VI.A.4.g (1[f]), VI.A.5 (2), VI.B.3 (4); EX1001, 117:12-24; EX1003, ¶157.

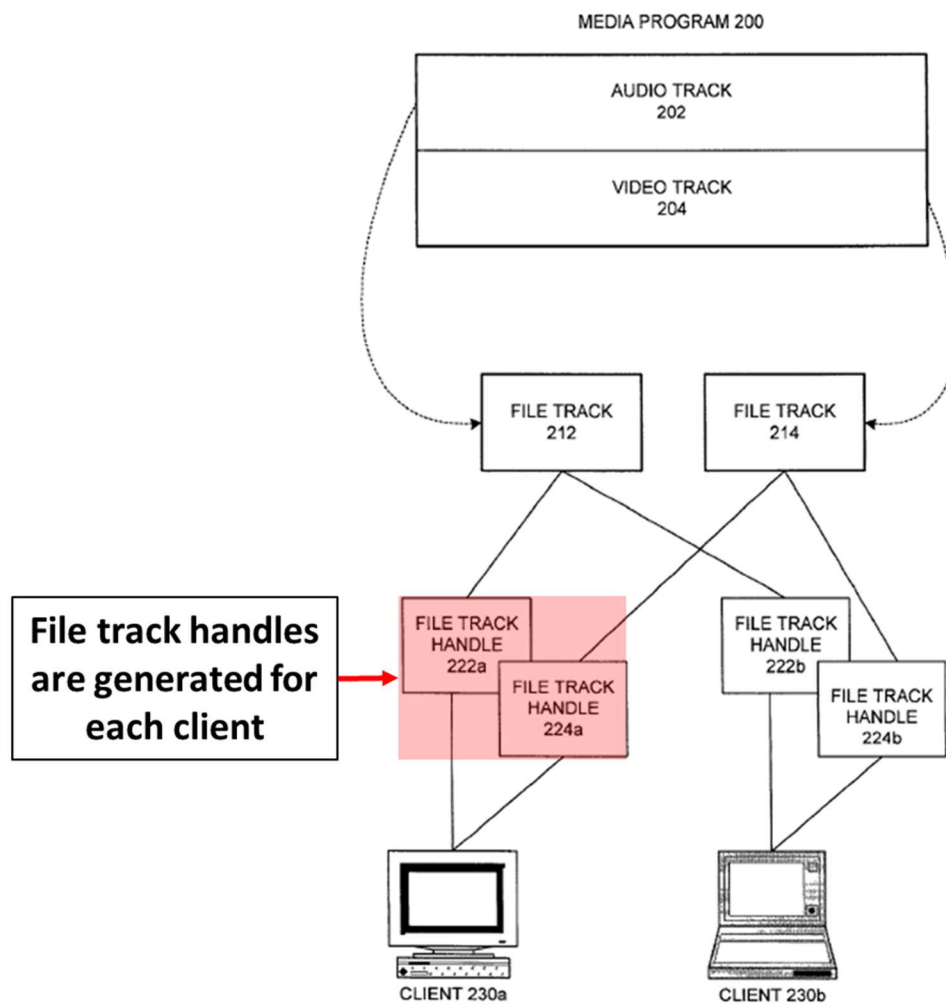
D. Ground 4: Claim 10 Is Obvious over *Bennett* in View of *Vadde*, *Riggs*, and *Srikantan*

1. Overview of *Srikantan*

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; EX1003, ¶158.



EX1055, FIG. 2.

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

A POSITA would have found it obvious 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; EX1003, ¶¶159-160. 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; EX1003, ¶159. 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* §VI.A.3; EX1003, ¶159. A POSITA would have recognized the bandwidth savings that would be gained from eliminating conflicting streaming sessions using *Srikantan*'s techniques. EX1003, ¶159.

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). *KSR*, 550 U.S. at 417; EX1003, ¶160. 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; EX1003, ¶160. 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; EX1003, ¶160. “[F]ile 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.” EX1003, ¶160; 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]

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” (“*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* §VI.A.4.e (1[d]). *Riggs* discloses that “the global address of content” (“*a network resource indicator that identifies the media object*”) is “typically” provided as a 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* §VI.A.6 (3); EX1003, ¶161.

b. 10[b]

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; EX1003, ¶162. 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*”). EX1003, ¶162.

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*, including “manag[ing] media connections” to include “Real-Time Transport Protocol (RTP)” sessions. EX1003, ¶163; EX1041, ¶¶25, 34, 54-55. 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. EX1003, ¶54. Similarly, *Srikantan* describes that its media servers use “RTP (Real-Time Transport Protocol) to deliver the stream to the client.” EX1055, ¶¶27, 46, 72. A POSITA would have recognized that when establishing an RTP session with a host server, the MA 206 would have retrieved a file handle 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; EX1003, ¶163.

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; EX1003, ¶164.

c. 10[c]

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* §VI.D.3.b (10[b]); EX1003, ¶165.

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; EX1003, ¶112. *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, which is a mode of operation consistent with a "***proxy***" as of the Critical Date. EX1056, 3-4; EX1057, 3 (describing that a "proxy" is "a device or program empowered to act for another"); EX1003, ¶166.

d. 10[d]

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; EX1003, ¶167. Additionally, “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; EX1003, ¶167.

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 §VI.D.3.b (10[b]); EX1003, ¶168. 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]

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; EX1003, ¶169.

VII. Mandatory Notices

A. Real Parties-in-Interest

Google LLC⁴ is a real party-in-interest.

Out of an abundance of caution, Google identifies Apple Inc. (“Apple”) as a real party-in-interest. Apple and Headwater have recently entered into “a worldwide settlement whereby Headwater licensed its patents to Apple.” Patent Owner’s Request for Discretionary Denial of Institution, *Google LLC v. Headwater Research LLC*, IPR2026-00048, Paper 8, 2 (PTAB Jan. 7, 2026).

As identified in the related matters below, the ’918 patent is being asserted against multiple Carriers in 2-25-cv-00906, 2-25-cv-00907, and 2-25-cv-00908. On December 1, 2025, the Carriers in those cases filed a joint opposed motion to stay the litigations under, among other things, the customer-suit exception. *See* Defendants’ Joint Opposed Motion to Stay, *Headwater Res. LLC v. DISH Network Corp.*, 2:25-cv-00907 (Dec. 1, 2025), ECF No. 52. On December 8, 2025, a redacted version of that motion was made publicly available. *See id.* ECF No. 63.

⁴ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

On December 19, 2025, Headwater filed a sealed response to the motion. *See id.* ECF No. 73. On January 9, 2026, the Carriers filed a sealed reply. *See id.* ECF No. 82. In light of these filings, Google names the following Carriers (the named defendants in these cases) as real parties-in-interest to this proceeding:

- DISH Network Corporation f/k/a EchoStar Communications Corporation
- DISH Network LLC
- Dish Wireless, LLC d/b/a Boost Mobile
- Charter Communications, Inc.
- Charter Communications, LLC
- Charter Communications Operating, LLC
- Comcast Cable Communications, LLC d/b/a Xfinity
- Comcast Cable Communications Management, LLC
- Comcast Corporation
- Comcast of Houston, LLC d/b/a Xfinity
- Spectrum Gulf Coast, LLC
- Spectrum Management Holding Company, LLC

To be clear, none of these Carriers funded, directed, or controlled the filing or preparation of this petition, nor did Google prepare or file the petition at the

Carriers' behest. Instead, Google makes this identification out of an abundance of caution and in the interest of over-disclosure.

B. Related Matters

The '918 patent is or has been involved in the following proceedings:

Name	Number	Forum	Filed
<i>Headwater Research LLC v. Apple Inc.</i>	7-25-cv-00379	W.D. Tex.	8/27/25
<i>Headwater Research LLC v. Google LLC</i>	7-25-cv-00374	W.D. Tex.	8/27/25
<i>Headwater Research LLC v. Dish Network Corp. et al</i>	2-25-cv-00907	E.D. Tex.	8/28/25
<i>Headwater Research LLC v. Charter Communications, Inc. et al</i>	2-25-cv-00906	E.D. Tex.	8/27/25
<i>Headwater Research LLC v. Comcast Cable Communications, LLC d/b/a Xfinity et al</i>	2-25-cv-00908	E.D. Tex.	8/28/25
<i>Headwater Research LLC v. Samsung Electronics Co., Ltd. et al</i>	2-23-cv-00641	E.D. Tex.	12/29/23
<i>Samsung Electronics Co., Ltd. v. Headwater Research LLC</i>	IPR2024-01397	P.T.A.B.	9/6/24
<i>Samsung Electronics Co., Ltd. v. Headwater Research LLC</i>	IPR2024-01396	P.T.A.B.	9/6/24

C. Lead and Back-Up Counsel

<u>Lead Counsel</u>	<u>Back-Up Counsel</u>
Erika H. Arner, Reg. No. 57,540 erika.arners@finnegan.com Finnegan, Henderson, Farabow, Garrett, & Dunner LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Phone: 571-203-2754 Fax: 202-408-4400	Joshua L. Goldberg, Reg. No. 59,369 joshua.goldberg@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-6092 Fax: 202-408-4400 Daniel C. Tucker, Reg. No. 62,781 daniel.tucker@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Phone: 571-203-2793 Fax: 202-408-4400 Benjamin A. Saidman, Reg. No. 69,325 benjamin.saidman@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 271 17th Street NW, Suite 1400 Atlanta, GA 30363-6209 Phone: 404-653-6510 Fax: 404-653-6444 Daniel F. Klodowski, Reg. No. 72,693 daniel.klodowski@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4216

<u>Lead Counsel</u>	<u>Back-Up Counsel</u>
	Fax: 202-408-4400 Elliot Cook, Reg No. 61,769 elliot.cook@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Phone: 571-203-2738 Fax: 202-408-4400 Michael V. Young, Reg. No. 61,180 michael.young@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Phone: 571-203-2788 Fax: 202-408-4400 Alexander M. Boyer, Reg. No. 66,599 alexander.boyer@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Phone: 571-203-2759 Fax: 202-408-4400 Pier D. DeRoo, Reg. No. 69,340 pier.deroo@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4418

<u>Lead Counsel</u>	<u>Back-Up Counsel</u>
	Fax: 202-408-4400 Christopher C. Johns, Reg. No. 68,664 Christopher.Johns@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4155 Fax: 202-408-4400 Kassandra M. Officer, Reg. No. 74,083 Kassandra.Officer@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4270 Fax: 202-408-4400 Bradford C. Schulz, Reg. No. 75,006 bradford.schulz@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Phone: 571-203-2739 Fax: 202-408-4400 William C. Neer, Reg. No. 78,874 william.neer@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4054

<u>Lead Counsel</u>	<u>Back-Up Counsel</u>
	Fax: 202-408-4400 Christopher B. Anderson, Reg. No. 77,898 christopher.anderson@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Phone: 571-203-2765 Fax: 202-408-4400 Kai Rajan, Reg. No. 70,110 kai.rajan@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4363 Fax: 202-408-4400 Umber Aggarwal, Reg. No. 76,098 umber.aggarwal@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Phone: 571-203-2423 Fax: 202-408-4400 Abhay A. Watwe, Reg. No. 59,461 abhay.watwe@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023

<u>Lead Counsel</u>	<u>Back-Up Counsel</u>
	Phone: 571-203-2716 Fax: 202-408-4400 Kathleen C. Galleher, Reg. No. 80,837 kathleen.galleher@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Phone: 571-203-2448 Fax: 202-408-4400 Connor M. McGregor, Reg. No. 79,634 connor.mcgregor@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4363 Fax: 202-408-4400 Heather J. Ren, Reg. No. 82,089 heather.ren@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4024 Fax: 202-408-4400 Yi Yang, <i>pro hac vice</i> to be filed yi.yang@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP

<u>Lead Counsel</u>	<u>Back-Up Counsel</u>
	2 Seaport Lane, 6 th Floor Boston, MA 02210 Phone: 617-646-1642 Fax: 202-408-4400

D. Service Information

Petitioner consents to being served electronically at the e-mail addresses provided above and at HeadwaterIPRTeam@finnegan.com.

VIII. Grounds for Standing

Petitioner certifies that the '918 patent is available for IPR and that Petitioner is not barred or estopped from requesting IPR challenging the patent claims on the grounds in this Petition.

IX. Conclusion

Petitioner requests IPR and cancellation of claims 1-19. The Office may charge any required fees to Deposit Account No. 06-0916.

Date: January 15, 2026

Respectfully submitted,

By: /Erika H. Arner/
Erika H. Arner, Lead Counsel
Reg. No. 57,540

CERTIFICATION OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, Petitioner certifies that the word count of Petitioner's Petition for *Inter Partes* Review (exclusive of any table of contents, table of authorities, mandatory notices under § 42.8, certificates of service and word count, and list of exhibits or claim listing) as measured by Microsoft Word is 13,979.

Dated: January 15, 2026

By: /Erika H. Arner/
Erika H. Arner, Lead Counsel
Reg. No. 57,540

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(a), the undersigned certifies that on January 15, 2026, a copy of the foregoing Petition for *Inter Partes* Review, the associated power of attorney, and Exhibits 1001-1050, 1053-1067 were served by FedEx Priority Overnight on the correspondence address of record indicated in the Patent Office's Patent Center website for U.S. Patent No. 9,647,918:

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

Date: January 15, 2026

By: /Lisa C. Hines/
Lisa C. Hines
Case Manager
FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, LLP