

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SPORTRADAR AG,
Petitioner

v.

SPORTSCASTR INC.,
(D/B/A PANDA INTERACTIVE),
Patent Owner.

IPR2025–00315
U.S. Patent No. 10,805,687

PETITION FOR *INTER PARTES* REVIEW UNDER 37 C.F.R. §42.101

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES UNDER 37 C.F.R. §42.8(A)(1)	2
	A. Real Parties–in–Interest.....	2
	B. Related Matters.....	2
	C. Lead and Back–up Counsel and Service Information.....	3
III.	REQUIREMENTS FOR IPR UNDER 37 C.F.R. §42.104	4
	A. Grounds for Standing	4
	B. Identification of Challenge and Relief Requested	4
	1. Claims for Which IPR is Requested	4
	2. Grounds for Challenge	4
	3. Claim Construction	6
	4. How the Claims Are Unpatentable	6
	5. Evidence Supporting This Challenge	6
IV.	TECHNOLOGY BACKGROUND: PROTOCOLS FOR REAL– TIME OR LOW–LATENCY DATA TRANSMISSION.....	7
V.	THE '687.....	7
	A. Alleged Problem.....	7
	B. Alleged Invention.....	7
VI.	LEVEL OF ORDINARY SKILL IN THE ART	9
VII.	OVERVIEW OF THE PRIMARY REFERENCES	10
VIII.	EACH OF THE CHALLENGED CLAIMS IS UNPATENTABLE	10
	A. Ground 1: Claim 1 is Obvious Over Ellis in View of Spivey.....	10

1.	Motivation to Combine	10
2.	Independent Claim 1	14
a.	Element 1.0	14
b.	Element 1.1	16
c.	Element 1.2	17
d.	Elements 1.3–1.4	18
i.	First VCD of the first/second plurality of VCDs	18
ii.	First copy of first/second plurality of copies	19
iii.	Providing copies via a first/second video Internet communication channel	19
e.	Element 1.5	20
f.	Element 1.6	24
g.	Element 1.7	28
h.	Element 1.8	29
i.	First event socket corresponding to the first event information germane to the first live sporting event	29
ii.	Transmitting the first event information to at least the first VCD of the first plurality of VCDs	32
iii.	Transmitting via a first event information Internet communication channel between at least one first event socket of the at least one socket server and the first VCD of the first plurality of VCDs	32
B.	Ground 2: Claim 10 is Obvious Over Ellis in View of Spivey and Kellicker	33

1.	Motivation to Combine	33
2.	Dependent Claim 10	35
a.	Elements 10.0–10.2	35
b.	Element 10.3	37
c.	Element 10.4	39
C.	Ground 3: Claim 11 and 15–16 are Obvious Over Ellis in View of Spivey, Kellicker and Phillips.....	40
1.	Motivation to Combine	40
2.	Dependent Claim 11	42
a.	Elements 11.0–11.2	42
b.	Element 11.3	46
c.	Element 11.4	46
3.	Dependent Claim 15	46
a.	Element 15.0	46
4.	Dependent Claim 16	48
a.	Element 16.0	48
D.	Ground 4: Claims 12–14 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips and Stoica.....	51
1.	Motivation to Combine	51
2.	Dependent Claim 12	54
a.	Elements 12.0–12.1	54
b.	Element 12.2	58
c.	Element 12.3	60
d.	Element 12.4	62

3.	Dependent Claim 13	62
a.	Elements 13.0–13.1	62
b.	Elements 13.2–13.3	63
4.	Dependent Claim 14	66
a.	Elements 14.0–14.1	66
b.	Elements 14.2–14.3	68
E.	Ground 5: Claims 17–18 are obvious Over Ellis in View of Spivey, Kellicker, Phillips and Pantos	70
1.	Motivation to Combine	70
2.	Dependent Claim 17	72
a.	Elements 17.0–17.1	72
b.	Element 17.2	72
c.	Element 17.3	72
d.	Element 17.4	72
e.	Element 17.5	72
i.	Playlist specifying different resolution versions	73
ii.	Chunklist pointers to respective chunklists	74
iii.	File segment pointers to respective file segments	75
f.	Element 17.6	77
g.	Element 17.7	78
h.	Element 17.8	78
3.	Claim 18	78

a.	Elements 18.0–18.1	78
b.	Element 18.2	79
c.	Element 18.3	79
d.	Element 18.4	79
IX.	SECONDARY CONSIDERATIONS OF NON–OBVIOUSNESS	80
X.	THE BOARD SHOULD NOT EXERCISE DISCRETIONARY DENIAL	81
A.	The '687 Has Not Been Subject to a Prior Petition by the Current Petitioner	81
B.	<i>Fintiv</i>	81
XI.	CONCLUSION	81

TABLE OF AUTHORITIES

Page(s)

Cases

<i>Geo. M. Martin v. Alliance Machine</i> , 618 F.3d 1294 (Fed. Cir. 2010)	80
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	6
<i>SportsCastr Inc. v. Genius Sports Ltd.</i> , No. 2:23-cv-04771-JRG (E.D. Tex.)	2
<i>SportsCastr Inc. v. Sportradar Group AG</i> , No. 2:23-cv-04772-JRG (E.D. Tex.)	2
<i>Videndum Production Solutions, Inc. v. Rotolight Limited</i> , IPR2023-01218, Paper 12 (April 19, 2024)	81

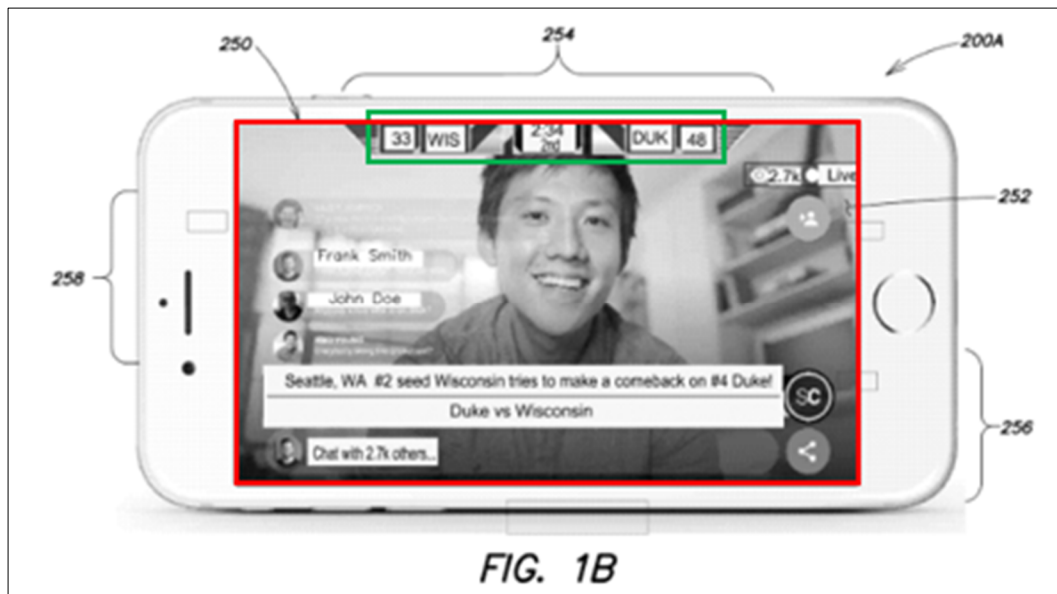
Other Authorities

37 C.F.R. §42.8(A)(1)	2
37 C.F.R. §42.10(b)	4
37 C.F.R. §42.63(d)(2)(i)	85
37 C.F.R. §42.100(b)	6
37 C.F.R. §42.104	4

I. INTRODUCTION

U.S. Patent No. 10,805,687 (“’687”) does not claim to have invented any new component. Rather, ‘687 is directed to the known concept of providing two or more related types of content to user devices over the Internet using known real-time communication protocols.

The ’687 provides a video feed (*i.e.*, a live stream of a sporting event) and at least one data feed (*e.g.*, the sporting event’s score) to each device over separate “communication channels.” Ex.1003, 26:48–59. The ’687’s final product is a **scoreboard feature**¹ displayed on top of a **video feed**:



Ex.1003, Fig. 1B (annotated). But displaying content, such as the score or game clock, has existed since at least 1994:

¹ *All emphases and coloring added, unless otherwise noted.



Ex.1045.

The '687 acknowledges that adapting this feature to live streaming and internet-based content distribution systems existed well before '687's earliest possible priority date of August 5, 2016. Ex.1003, 2:4–27. Thus, for the reasons detailed in the asserted grounds below, Petitioner respectfully requests *inter partes* review ("IPR") of the Challenged Claims.

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

A. Real Parties-in-Interest

Sportradar AG ("Petitioner"), Sportradar Group AG, Sportradar Solutions LLC and Sportradar US LLC are the real parties-in-interest.

B. Related Matters

Patent Owner ("PO") asserted '687 against Petitioner in *SportsCastr Inc. v. Sportradar Group, AG.*, No. 2:23-cv-04772-JRG (E.D. Tex.). PO also asserted '687 against Genius Sports Ltd. ("Genius") in *SportsCastr Inc. v. Genius Sports Ltd.*,

No. 2:23-cv-04771-JRG (E.D. Tex.). Both cases (“Texas Litigation”) were filed October 5, 2023. The ’687 is one member of a patent family, other members of which are also asserted in the Texas Litigation, and each of which shares the same specification: U.S. Patent Nos. 10,805,687; 10,425,697; and 11,039,218. Exs.1003–1011. The ’687 is the subject of IPR Petitions filed by Genius (“Genius IPRs”): IPR2024–01305 (claims 1–9) and IPR2024–01307 (claims 19–25, 27–30); ’697 is the subject of IPR2024–01308 (claims 1–9) and IPR2024–01309 (claims 19–25, 27–30); and the ’218 is the subject of IPR2024–2024–01310 (claims 1–15) and IPR2024–01311 (claims 16–30). Petitioner also intends to seek joinder with the Genius IPRs. *Id.*

The current petition is directed to claims that were previously unasserted by PO in the Texas Litigation, and thus were not the subject of the Genius IPRs. Exs.1014, 1048. This case may affect, or be affected by, the Texas Litigation and the Genius IPRs.

C. Lead and Back-up Counsel and Service Information

Lead Counsel	Backup Counsel
Scott W. Cummings Reg. No. 41,567 scott.cummings@bclplaw.com BRYAN CAVE LEIGHTON PAISNER LLP 1155 F St, N.W. Washington, D.C. 20004 Telephone: (202) 508–6000	Alexander Walden provisionally recognized PTAB attorney under 37 C.F.R. §42.10 (<i>pro hac vice</i> affidavit to follow) alexander.walden@bclplaw.com

	Frank Fabiani Reg. No. 75,934 frank.fabiani@bclplaw.com BRYAN CAVE LEIGHTON PAISNER LLP 1290 Avenue of the Americas New York, NY 1104 Telephone: (212) 541–2000 Facsimile: (212) 541–4630
--	--

A Power of Attorney accompanies this Petition pursuant to 37 C.F.R. §42.10(b). Petitioner consents to electronic service by email at scott.cummings@bclplaw.com, alexander.walden@bclplaw.com, frank.fabiani@bclplaw.com, and SportradarIPR@bclplaw.com.

III. REQUIREMENTS FOR IPR UNDER 37 C.F.R. §42.104

A. Grounds for Standing

Petitioner certifies ‘687 is available for IPR and that Petitioner is not barred or estopped from requesting IPR on the grounds identified herein.

B. Identification of Challenge and Relief Requested

1. Claims for Which IPR is Requested

Petitioner challenges claims 1 and 10–18 of ‘687.

2. Grounds for Challenge

Petitioner challenges the claims based on the following references:

1. U.S. Patent Publication No. 2014/0229992 to Ellis *et al.* (“Ellis”) published August 14, 2014;

2. U.S. Patent Publication No. 2016/0036910 to Spivey (“Spivey”) published February 4, 2016;

3. U.S. Patent Publication No. 2017/0250882 to Kellicker (“Kellicker”), filed on February 29, 2016, and published in August 31, 2017;

4. U.S. Patent Publication No. 2015/0127845 to Phillips (“Phillips”) published May 7, 2015;

5. U.S. Patent No. 9,246,965 to Stoica *et al.* (“Stoica”) published January 26, 2016; and

6. R. Pantos *et al.*, “HTTP Live Streaming,” (“Pantos”) published April 4, 2016. Ex.1053.

None of this prior art was applied during prosecution. *See generally* Ex.1005.

Petitioner requests IPR on the following grounds:

Ground	Basis	Claim(s)	References
1	§103	1	Ellis in view of Spivey
2	§103	10	Ellis in view of Spivey and Kellicker
3	§103	11, 15–16	Ellis in view of Spivey, Kellicker and Phillips
4	§103	12–14	Ellis in view of Spivey, Kellicker, Phillips and Stoica
5	§103	17–18	Ellis in view of Spivey, Kellicker, Phillips and Pantos

3. Claim Construction

Claims are construed according to the principles set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*). 37 C.F.R. §42.100(b). The claim terms of '687 should be given their plain and ordinary meaning as understood by a person of skill in the art ("POSA").

The challenged claims are system claims. However, many of the elements recited in the claims constitute nothing more than recitations of intended use and/or functionality and specific types of content/data, none of which serves to further limit the claimed system. Petitioner expressly reserves the right to challenge the limiting effect of any such claim element.

The claim constructions proposed for purposes of this proceeding are not a waiver of any argument in any litigation that claim terms in '687 are indefinite or otherwise invalid or unpatentable.

4. How the Claims Are Unpatentable

§VIII. *infra*, explains how the Challenged Claims are unpatentable.

5. Evidence Supporting This Challenge

A list of exhibits is provided at the end of the Petition. The relevance of this evidence and the specific portions supporting the challenge are provided in this Petition and in Ex.1001. Petitioner relies on, and incorporates herein, the entirety of Ex.1001 in support of this Petition and reserves the right to rely on all of Ex.1001 in response to arguments raised by PO.

IV. TECHNOLOGY BACKGROUND: PROTOCOLS FOR REAL-TIME OR LOW-LATENCY DATA TRANSMISSION

An overview of conventional protocols, including HTTP Polling and Websocket, is set forth in the accompanying declaration. Ex.1001 ¶¶31–59.

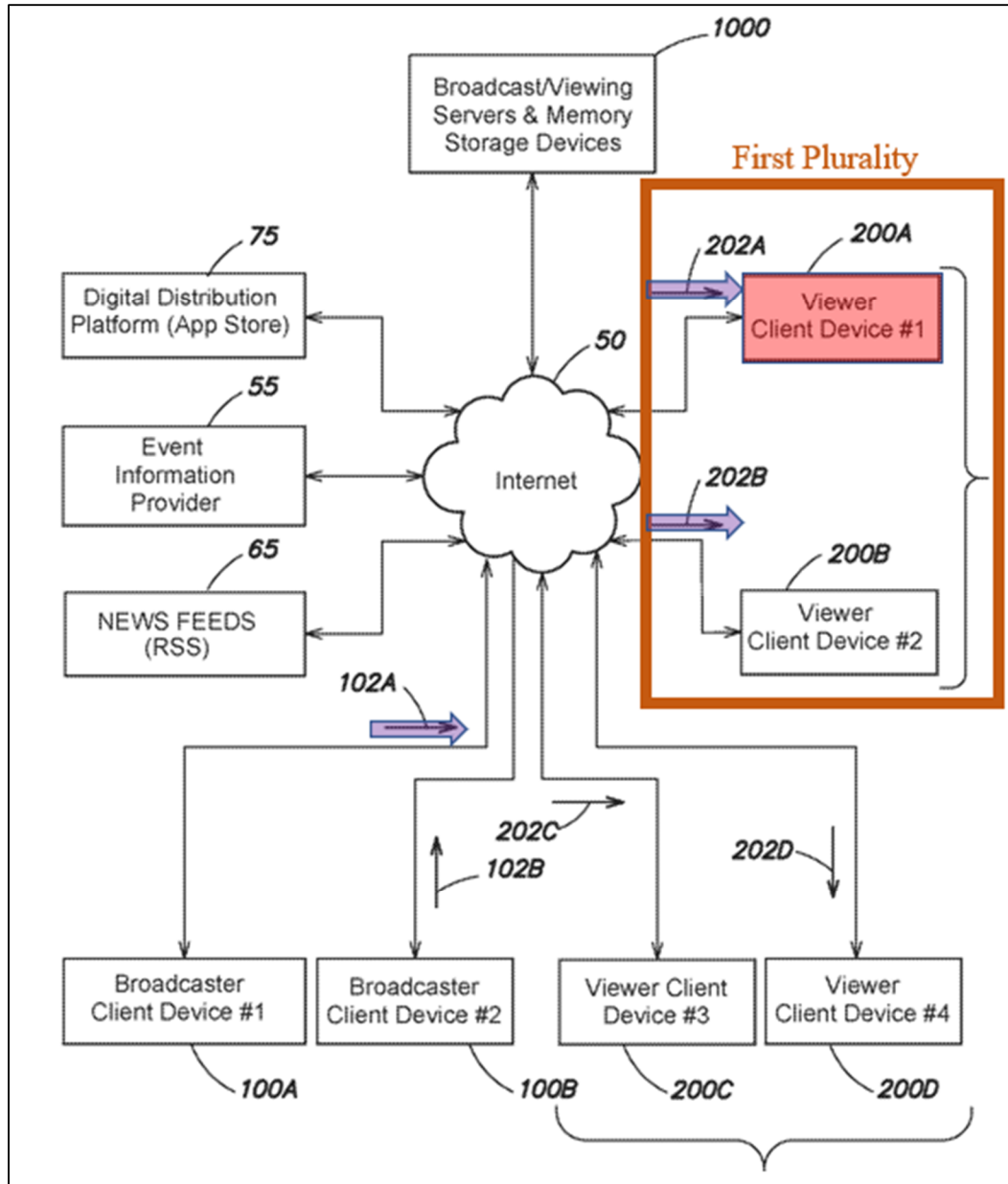
V. THE '687

A. Alleged Problem

According to the '687, prior art techniques for streaming video to multiple users suffered from a latency problem. Ex.1003, 2:31–40, 5:16–37; Ex.1001 ¶¶60–62.

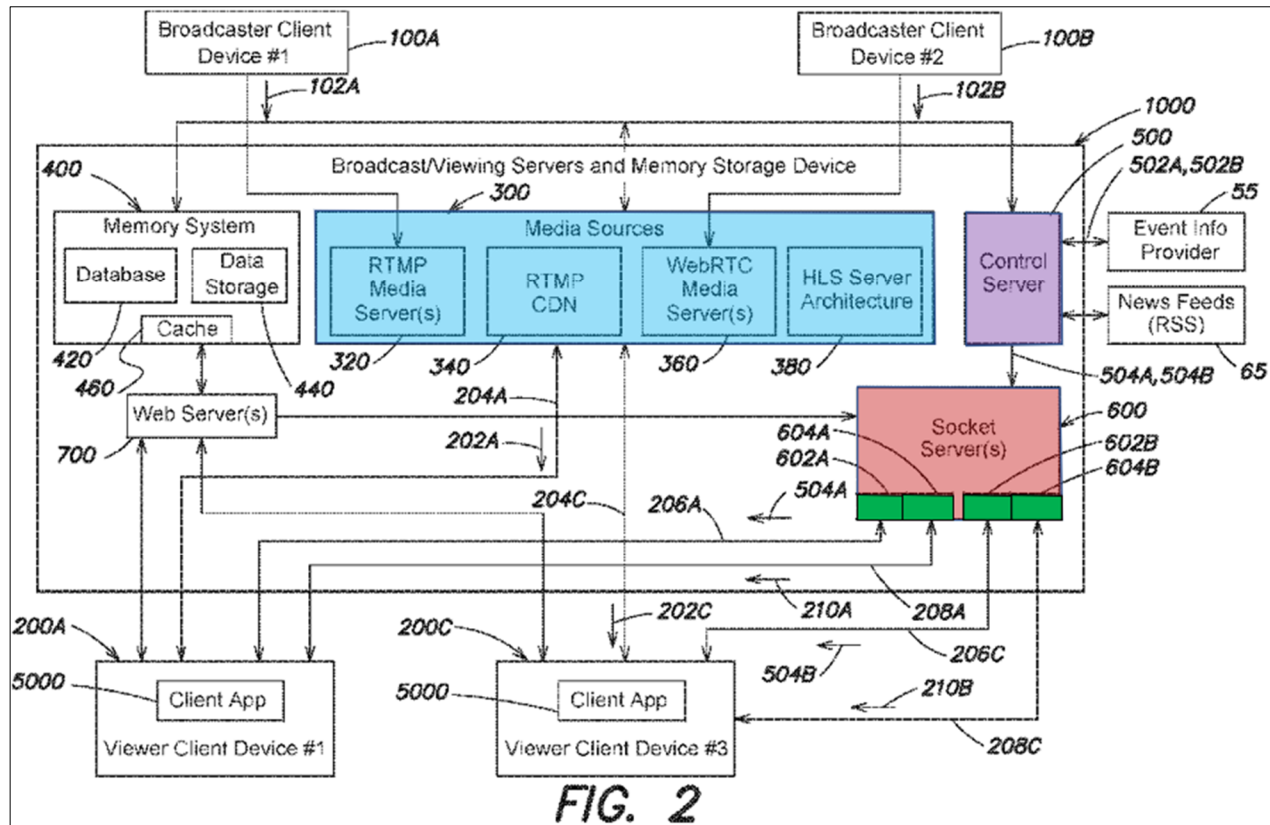
B. Alleged Invention

The '687 is directed to a “server and memory storage architecture” for obtaining and providing to viewers, (1) an “audio and/or visual feed” including “commentary associated with [a] live event” and (2) a “separate event data feed including real time information associated with the ... live event.” Ex.1003 7:16–37. This architecture is shown in Figures 1A and 2 (reproduced below) and discussed in more detail in the accompanying expert declaration. Ex.1001 ¶¶63–71.



Id. Fig. 1A(annotated), 11:19–22. Figure 1A depicts providing “a first viewer client device (200A)” of “a first plurality of viewer client devices (200A, 200B)” individual “copies (202A, 202B) of a first broadcaster’s live stream of digital content (102A).” *Id.* 9:25–50.

Figure 2 depicts viewer client devices (“VCDs”), media sources, a control server, a socket server, sockets, an event information provider, and a memory system.



Id. Fig. 2 (annotated).

VI. LEVEL OF ORDINARY SKILL IN THE ART

A POSA in the field of live streaming technology at the time of the alleged invention, August 5, 2016, would have had at least a bachelor’s degree, or equivalent, in electrical engineering, computer engineering, computer science, or a related field, and at least two years of experience in video and content processing and communications systems and methods. Ex.1001 ¶72–76. Additional education

or experience in related fields could compensate for deficits in the above qualifications. *Id.*

VII. OVERVIEW OF THE PRIMARY REFERENCES

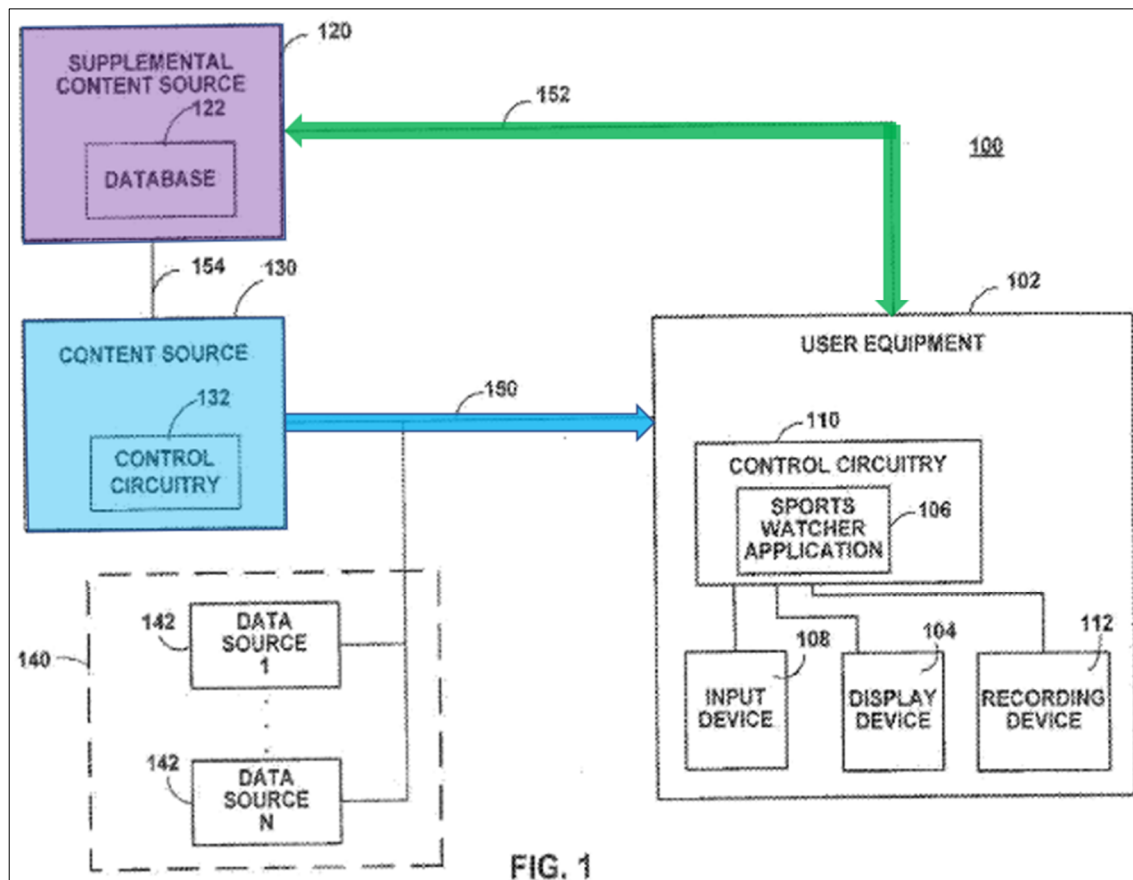
An overview of the cited references is set forth in the accompanying declaration. See Ex.1001 ¶¶79–104.

VIII. EACH OF THE CHALLENGED CLAIMS IS UNPATENTABLE

A. Ground 1: Claim 1 is Obvious Over Ellis in View of Spivey

1. Motivation to Combine

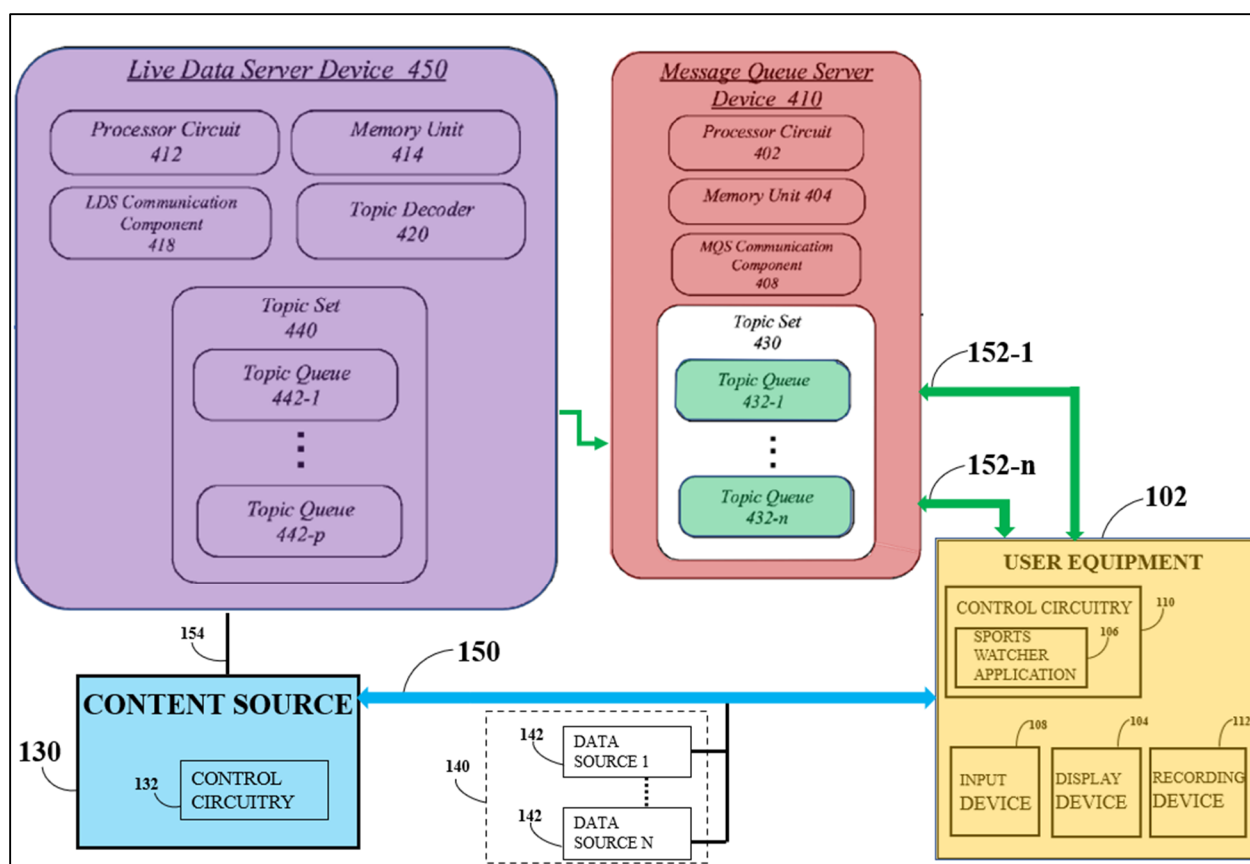
Ellis discloses ‘687’s broad functionality—providing (1) video and (2) data to a user over separate connections:



Ex.1016 Fig. 1(annotated); Ex.1001 ¶¶105. Spivey provides details relating to backend server architecture supporting the described functionality and the use of a websocket connection in lieu of HTTP to transmit Ellis’s SCS information. Ex.1001 ¶¶106–07.

Spivey explains that some live–event data delivery systems, such as Ellis’s, suffer from latency issues. Ex.1001 ¶106. Spivey teaches a Live Data Server Device (“LDSD”) and Message Queue Server Device (“MQSD”) to provide live–event data to any number of customers in milliseconds. *Id.*; Ex.1017 ¶¶27–28. A POSA, looking to improve Ellis’s latency issues, would have implemented a system such as Spivey, and would have recognized that combining Spivey’s teachings with Ellis would improve Ellis’s latency issues. Ex.1001 ¶¶107–09, 93–104.

A POSA would have modified Ellis’s **data** delivery system by replacing Ellis’s **SCS** with Spivey’s **LDSD** and adding the **MQSD** to achieve a predictable result of decreased latency data delivery using a websocket connection in lieu of HTTPS. *See* Ex.1016 ¶¶39, 41, Fig. 1; Ex.1017 ¶¶53–57, Fig. 4; Ex.1001 ¶¶110. An example of this combination is illustrated here in an annotated, combined diagram of Ellis Figure 1 and Spivey Figure 4:



See Ex.1016 Fig. 1; Ex.1017 Fig. 4; Ex.1001 ¶110. The combined system would maintain Ellis’s video content delivery system and upgrade the data delivery system to reduce latency. Ex.1001 ¶110.

A POSA would have been motivated to combine Ellis and Spivey. First, both relate to the same subject matter—content delivery for real-time events—and both describe streaming video and/or event data to one or more users simultaneously. See Ex.1016 Abstract, ¶156; Ex.1017 Abstract; Ex.1001 ¶¶111–12. Second, the references solve the same or similar problems, namely, quickly providing video and/or event data to multiple users over the Internet. See, e.g., Ex.1016 ¶¶39–40, 176; Ex.1017 ¶27. Ex.1001 ¶112.

A POSA would have successfully combined Ellis and Spivey. Ellis and Spivey disclose complementary means of achieving the common goal of synchronizing multiple forms of data, including video/audio and event data and transmitting that data to one or more users. For example, Ellis discloses various systems for transmitting one or more live event videos and related event data to multiple viewers over the Internet. *See, e.g.*, Ex.1016 ¶¶39–41, 45, 74; Ex.1001 ¶112.

A POSA also would have been motivated to combine Ellis with Spivey because Spivey provides a complementary, detailed solution for the backend architecture for transmitting event data to multiple users. Ex.1017 Abstract; Ex.1001 ¶114. For example, to implement Ellis’s system and reduce its latency, a POSA would have had a reasonable expectation of success in using Spivey’s system of “topic queues and open duplex communication connections” employing websockets and a “message queue server device” to broadcast “data from a live event (*e.g.*, news event, a gaming event, sporting event, etc.)” to multiple users “in real–time.” Ex.1017 ¶¶27, 58.

Implementing certain portions of Spivey’s system into Ellis’s system would have required only the simple substitution of known elements to obtain predictable results. Ex.1001 ¶113. For example, Ellis discloses compatible and customizable hardware and software components to act as content/data sources for event

information to be transmitted to users. *See, e.g.*, Ex.1016 ¶47. Predictably, therefore, a POSA would have simply substituted portions of Ellis’s SCS with Spivey’s LDSD and MQSD. ¶. For example, it would have been obvious to substitute any of Ellis’s elements with any combination of Spivey’s LDSD, MQSD, and/or Live Data Source(s), which are similarly involved in transmitting live event data to users and may be configured in any manner. *See* Ex.1017 ¶34; Ex.1001 ¶113.

2. Independent Claim 1

a. Element 1.0²

To the extent the preamble is limiting, Ellis discloses this element. *See* Ex.1016 Abstract. Ellis discloses a system for providing digital content relating to a live sporting event to a first/second plurality of VCDs by providing “enhanced information access and viewing functionality for sports–related programming,” which includes “a sporting event, sports news, or sports commentary.” *Id.* ¶¶8, 11, 38; Ex.1001 ¶432. Figure 2B, below, shows sporting events available at the same time to any number of users.

² Ex.1004 is a claim listing that enumerates each claim element.

220		230		
222	224	232	234	236
Tuner	Primary Source ID	Source ID	Type	Profile Score
1	<sourceid_2>	<sourceid_1>	Game	90
2	<sourceid_1>	<sourceid_2>	Game	95
3	<sourceid_8>	<sourceid_3>	Commentary	45
4	<sourceid_4>	<sourceid_4>	News	75
		<sourceid_5>	Movie	25
		<sourceid_6>	Movie	35
		<sourceid_7>	Highlights	65
		<sourceid_8>	Game	80

Id. ¶¶39, 62–63, 74, Fig. 2B(showing multiple live broadcasts available).

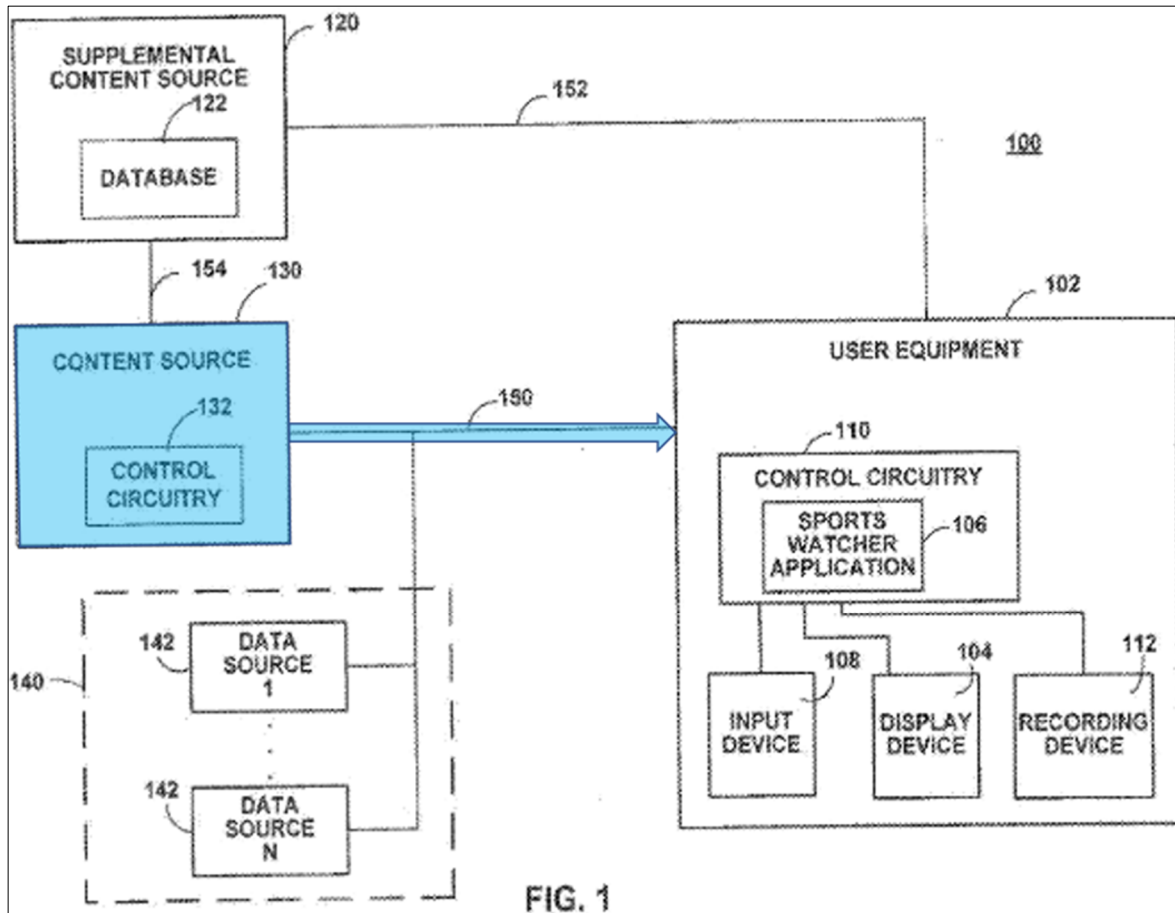
Ellis discloses a system that provides the “media content” from sporting events “to user equipment 102.” *Id.* ¶¶38–40. Ellis defines media content as broadcast programming “and any other content capable of being displayed by, presented to, ... or interacted with, using user equipment 102.” *Id.*

Ellis discloses multiple broadcasters providing copies of each broadcaster’s live stream of digital content including content relating to the live sporting event to a plurality of user devices by providing a broadcast system where any “number of users” may receive signals from a content source. Ex.1016 ¶39. The signals are individual copies of the same live stream of digital content. *Id.* ¶¶40, 62, Fig. 2B; Ex.1001 ¶433. For example, Ellis’s system broadcasts digital streams from the content source, which can be an “[i]nternet or network media/web server.” Ex.1016 ¶40. Users can select which content to receive, as shown in Figure 2. *Id.* ¶62, 74. Ellis then sends an individual copy of the live event to the users over a

communication path—any “suitable wired or wireless path.” *Id.* ¶¶45, 74. “For IPTV,” content may be “delivered on a particular network address (*e.g.*, multicast address) on which an IP socket is currently joined.” *Id.* ¶74. A POSA would have understood the distribution system of Ellis as providing a plurality of copies of multiple broadcaster’s live stream of digital content to a plurality of viewers. Ex.1001 ¶433.

b. Element 1.1

Ellis discloses **content sources**, which are “any suitable source of media content” such as a “Internet or network media/web server, or any other suitable facility or system for originating or distributing passive or interactive content to user equipment.” Ex.1016 ¶¶39–40. Ellis’s content sources receive the first/second broadcaster’s live stream of digital content and distribute video content to users via communications paths. *See id.* ¶¶39–41, 45, 47; *see also* Ex.1001 ¶434. Figure 1 shows a content source distributing video content to users over communication path 150:



Ex.1016 ¶¶39–41, Fig. 1(annotated). Ellis also discloses multiple content sources, *i.e.*, a plurality. *Id.* ¶39, 74(“[T]here may be numerous instances of content source 130”); Ex.1001 ¶434. Ellis further discloses that users can be connected with a plurality of data sources 140, 142. Ex.1016 Fig. 1. Moreover, as explained below, Ellis’s content sources perform the same functions as the media sources recited in elements 1.2–1.4.

c. Element 1.2

Ellis discloses that the content source “may be any suitable source of media content, such as, for example a ... television broadcast facility, ... Internet or

network media/web server, or any other suitable facility or system for ... distributing ... content to user equipment 102.” *Id.* ¶40. The content sources provide multiple streams of digital content, including “broadcast programming” and live sporting events. *Id.* ¶¶40, 45, 74. Ellis further discloses that a user may “actively watch one or more sporting events,” and the user “may change or swap primary and/or secondary sporting events at any time.” *Id.* ¶¶61–62. Thus, Ellis discloses content sources that receive multiple broadcaster’s live streams of digital content. Ex.1001 ¶435.

d. Elements 1.3–1.4

i. First VCD of the first/second plurality of VCDs

Ellis discloses a first VCD of a first/second plurality of VCDs because Ellis discloses an “interactive media system” where “any suitable number of users” may “select any game from the listing” of “available games” to “watch the selected game.” Ex.1016 ¶39, 94. For example, a first user in a first plurality of users can choose to watch the Yankees broadcast from the YES network. *Id.* ¶55; Ex.1001 ¶¶436–37. Meanwhile, a first user of a second plurality of users can choose to watch a separate live stream of digital content, such as a different game or game commentary. Ex.1016 ¶¶61–62, 132, Fig. 2B; Ex.1001 ¶437. Thus, Ellis discloses multiple first users of multiple different pluralities of VCDs. Ex.1001 ¶437.

ii. First copy of first/second plurality of copies

Ellis discloses a first copy of a first/second plurality of copies because Ellis discloses an “interactive media system” where “any suitable number of users” may “select any game from the listing” of “available games.” Ex.1016 ¶¶39, 94. In other words, Ellis discloses a system that provides live streams of digital content from multiple live games as individual copies sent to individual users. *Id.* ¶¶39, 61–62, Fig. 2B; Ex.1001 ¶438–439. Ellis’s content source, an internet server, provides the live stream over any “suitable wired or wireless path” including an internet connection. Ex.1016 ¶¶39, 45; Ex.1001 ¶439. Thus, each user would receive an individual copy of the live stream of digital content. Ex.1001 ¶439.

iii. Providing copies via a first/second video Internet communication channel

Ellis discloses providing individual copies of multiple different live streams to multiple groups of client devices. §VIII.A.2.a, d.; Ex.1001 ¶440. Ellis further discloses providing digital content or video content over a video Internet communication channel. As an example of an “Internet communication channel,” ‘687 describes a “video” channel between a media server and a client device to convey video–based commentary. Ex.1003, 5:45–50. Ellis discloses providing video from a content source (*e.g.*, “internet or network media/web server”) to user equipment (*e.g.*, computer equipment), over a communication path (“for example, a satellite path, a fiber–optic path, a cable path, or any other suitable wired or wireless

path.”). Ex.1016 ¶¶39–40, 45, 48, 74. Ellis describes the communication path as a unique internet connection between the content source and the user equipment that provides a video live stream: “[f]or IPTV, tuned content may include content to be delivered on a particular network address (*e.g.*, multicast address) on which an IP socket is currently joined.” *Id.* ¶74. Thus, Ellis discloses that each user receives a copy from the content source over a video internet communication path that is unique to each user device. Ex.1001 ¶440.

e. Element 1.5

To the extent that the frequency with which the control server retrieves information is considered as further limiting the claimed system, Ellis and Spivey both disclose this element. Ex.1001 ¶441. The ’687 explains that the **control server** is a generic server that retrieves “event information” from an “event information provider” and transmits it to the socket server. Ex.1003, 55:15–25, 19:51–55, 21:46–62. The event information is “germane to” a particular “live sporting event,” *e.g.*, score information of one of the live streams of digital content. *Id.* 19:37–58, 21:5–46.

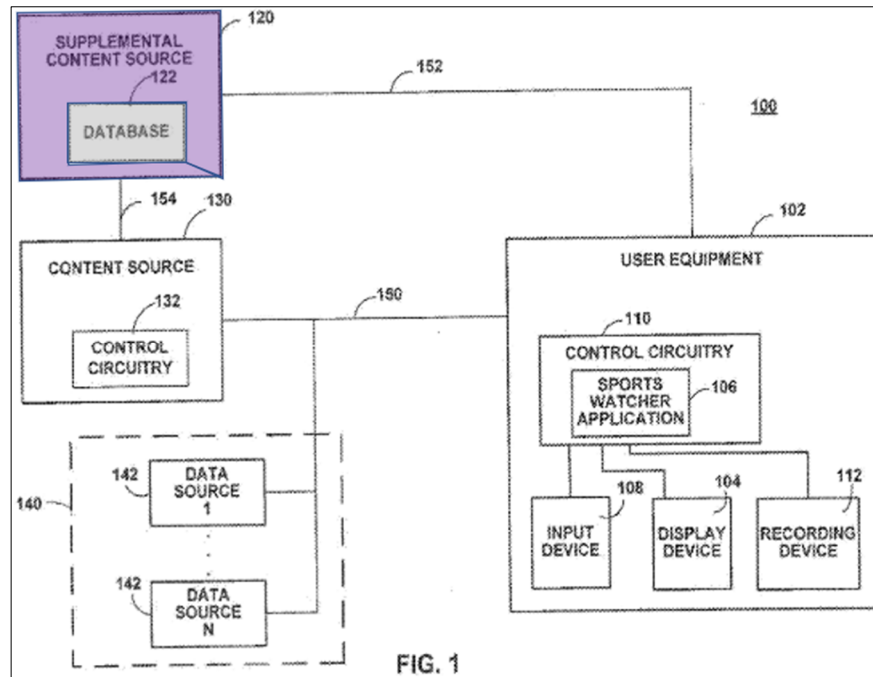
Ellis’s **SCS** comprises a control server. Ex.1016 ¶45(“supplemental content source 120 may be configured to transmit signals to user's equipment 102 over any suitable communication paths”). The SCS can be “any suitable content source configured to provide real–time sports scores and game summary information”

including a “content manager, content aggregator, or intermediate content provider.” *Id.*; Ex.1016 ¶¶42, 142. The SCS periodically retrieves, via the internet and from an event information provider, first event information germane to the first live sporting event, including first score information for the first live sporting event because the SCS “provide[s] real–time sports scores and game summary information.” *Id.* ¶¶41, 72, 117; Ex.1001 ¶442. A POSA would have understood that this information is retrieved via the internet. *Id.* ¶¶37–38, 40, 142. Furthermore, that is how real–time data was transferred to users at the time of Ellis and Spivey—including score updates and wagering information. Ex.1016 ¶¶42, 142; Ex.1001 ¶442. Additionally, it would have been obvious to a POSA to have retrieved this information via the Internet as a well–known, convenient and reliable means of acquiring the supplemental content. *Id.*

While a “periodic” time interval is not specifically mentioned in Ellis, a POSA would have understood that any system that provides real–time sports statistics, such as Ellis, would update the statistics on a periodic basis. Ex.1001 ¶443. This is especially true given that no specific time interval is specified by the claim.

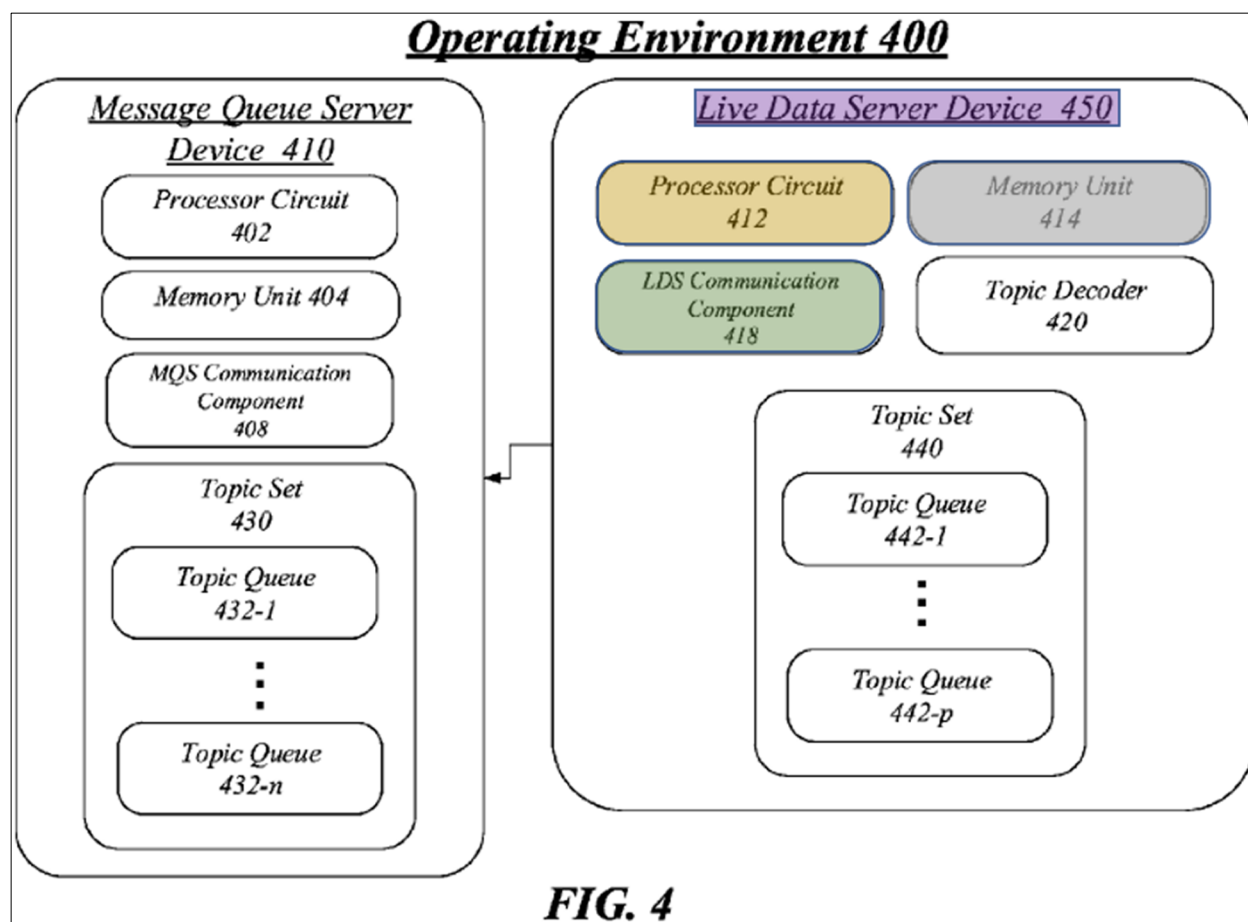
Alternatively, it would have been obvious to a POSA to implement periodic retrieval of the event information in order to provide the user with up–to–date information thereby enhancing the overall viewing experience, whilst limiting queries to the system and load on the components. Ex.1001 ¶444.

Further, as depicted in Figure 1, the SCS may store the received “sports–related information ... in database 122 and provided to content source 130 or directly to user equipment 120 over communication paths 154 and 152 . . .”:



Ex.1016 ¶42, Fig. 1(annotated).

To the extent Ellis does not expressly disclose or render obvious this element, Spivey’s [LDSD](#), as shown in Figure 4, contains the same basic components as ‘687’s [control server](#):



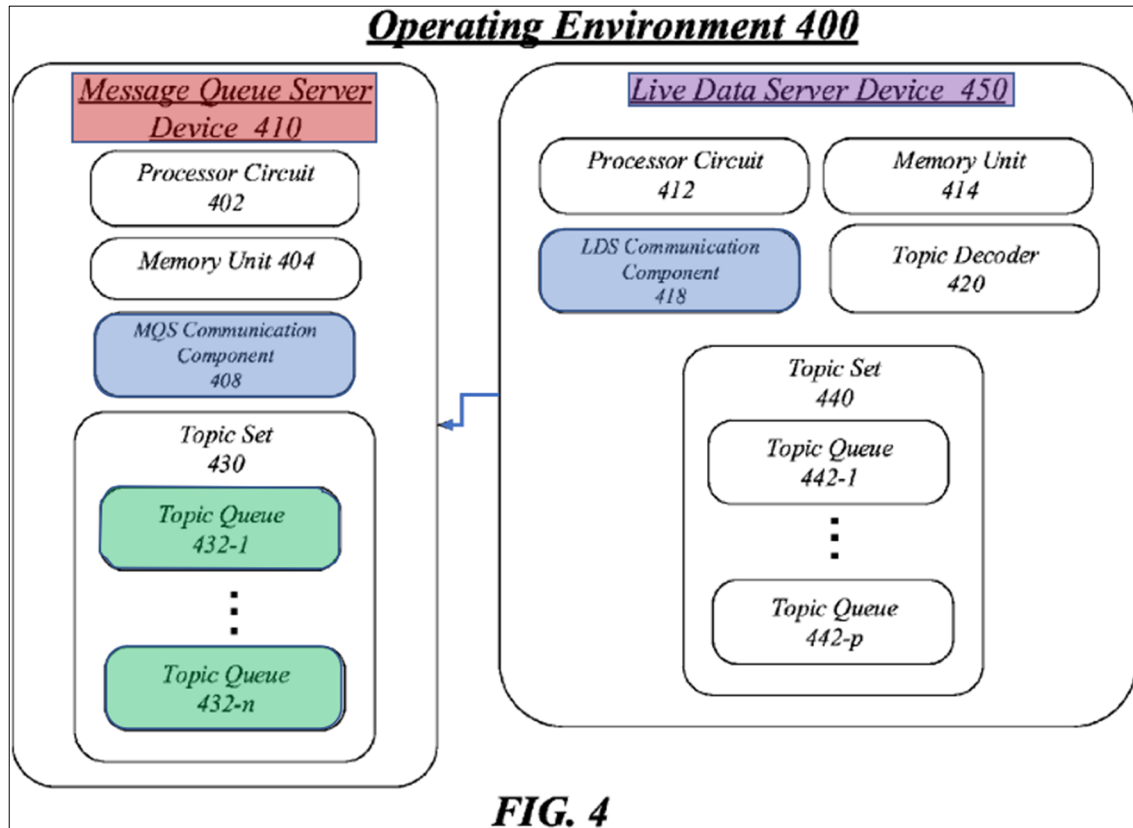
Ex.1017 ¶¶54–55; Fig. 4(annotated). The LDS is a generic server comprising one or more **processors**, memory devices, and **communication interfaces**. *Id.*; Ex.1001 ¶445. Moreover, the LDS performs the same functions as ‘687’s control server. Ex.1001 ¶446. The LDS contains an LDS communication component “to receive live data and transmit the data payloads to MQS device 410.” Ex.1017 ¶55. The live data received by the LDS includes “sports scores and breaking news,” such as “a score, a play, an error, ... and so forth.” *See id.* ¶¶27, 48, 66(“A data payload may include ... information about the live event in progress”); Ex.1001 ¶446. The LDS retrieves the first event information germane to a first live sporting event including

the first score information for the first live sporting event via the internet and from an event information provider because the live data sources are associated with the live event, and servers in Spivey communicate over the internet. Ex.1017 ¶¶35, 37, 178; Ex.1001 ¶446. Thus, a POSA would have understood Ellis and Spivey both disclose this element. Ex.1001 ¶446. To the extent the SCS does not meet this element, a POSA would have replaced it with Spivey's LDS and MQSD to decrease latency. *Id.*; §VIII.A.1.

f. Element 1.6

Ellis in view of Spivey discloses a **socket server** communicatively coupled to the **control server** (disclosed by Ellis's **SCS** and Spivey's **LDS**). The '687 describes the socket server as a generic server that "facilitate[s] communication" of "synchronized event information retrieved by the control server" associated with an event between the control server and the VCDs. *See* Ex.1003, 55:15–25, 20:7–21. The socket server does so by using "one or more **sockets** ... dedicated to the particular event" allowing VCDs "to establish an event information channel with the socket server." *Id.* 20:5–11. The socket server then receives event information transferred from the control server and shares it in "event messages" via the event information channel "in a synchronized manner" to all viewers "following the particular event." *Id.*; Ex.1001 ¶447.

Ellis discloses providing event information over a separate connection to user equipment. Ex.1016 ¶¶39–41, Fig. 1. A POSA would have looked to Spivey to add additional backend architecture to reduce latency in Ellis’s real-time data delivery system. Ex.1001 ¶¶448–49. Like ‘687’s socket server, Spivey’s **MQSD** is a generic server that contains processors, memory, and communication components. Ex.1017 ¶¶35, 53, Figs. 3, 4. Moreover, Spivey’s MQSD carries out the same purpose as ‘687’s socket server. The MQSD “receive[s]” and “process[es] live data to provide live data” to “the subscribed connected client devices substantially simultaneously.” Ex.1017 ¶96; *id.* ¶¶45(“[MQS] device 310 may be a representative example of message queue server 120”), 54(MQSD is “substantially similar” to “MQS device 310”). To accomplish this, the MQSD receives live event data from the LDSD, which just “receive[d] live data:” Ex.1001 ¶449.



Id. ¶54, 55, Fig. 4(annotated). The MQSD is communicatively coupled to the LDSD through the MQS communication component and the LDS communication component. *Id.* ¶46 (communication component 308 transmits and receives data), 54 (communication component 408 is analogous to 308), 55 (“LDS communication component 418 ... allows [LDSD] to ... transmit the data payloads to [MQSD].”).³

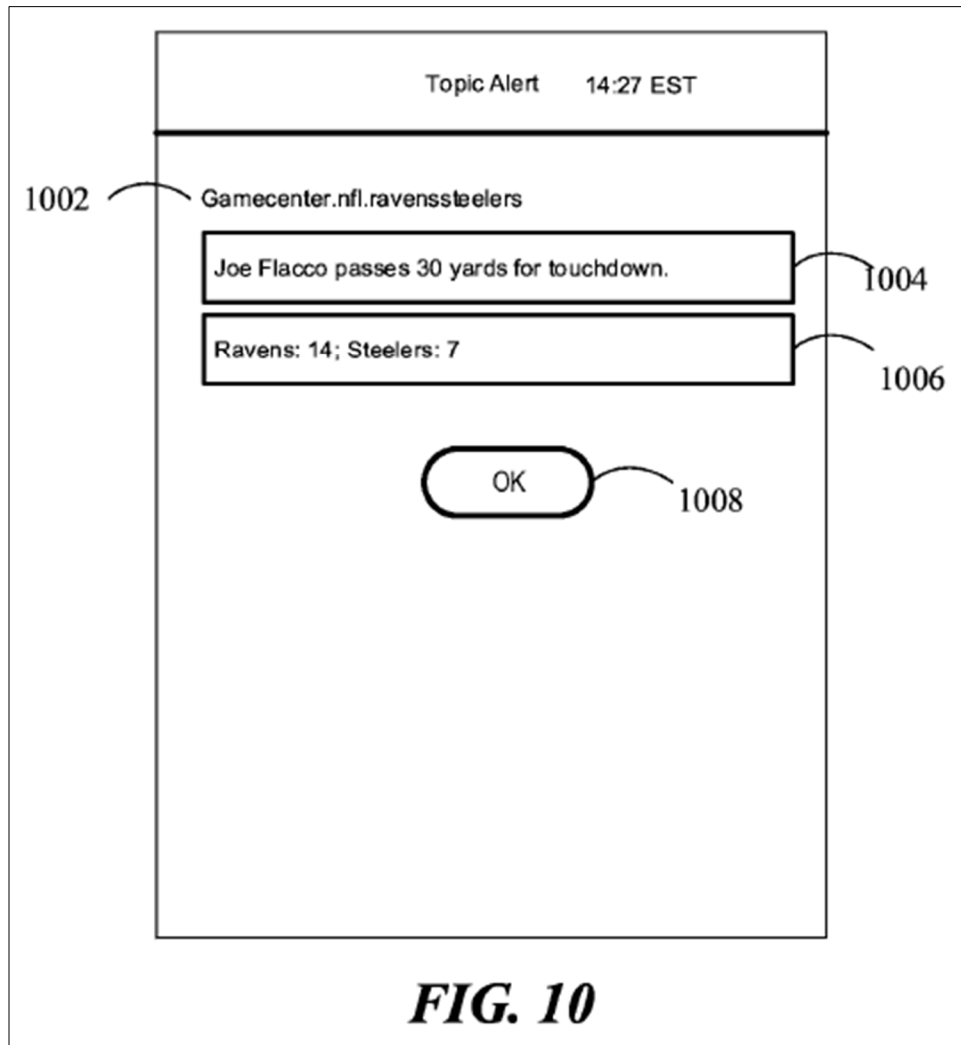
³ Ellis also discloses a system where multiple servers provide the event information. Ex.1016 ¶47 (SCS’s functionality “may be provided by multiple systems at multiple facilities”).

The MQSD organizes the received live data “into one or more of a plurality of **topic queues**” dedicated to storing information about “a specific game.” *Id.* ¶¶47, 52. The MQSD then shares this information “substantially simultaneously” to “the subscribed connected client devices.” *See id.* ¶96. Thus, a POSA would have understood Spivey’s MQSD as disclosing a socket server communicatively coupled to a control server. Ex.1001 ¶449.

Alternatively, Spivey teaches that latency associated with the broadcast of event data, such as sports scores, to clients can be reduced by “using topic queues and open duplex communication connections.” Ex.1017 ¶¶27–28. Here, “topics” include specific sporting events. *Id.* ¶85 (“topic 910–1 corresponding to a National Football League (NFL) game between the Baltimore Ravens and the Pittsburgh Steelers.”). Spivey discloses that such “open duplex communication connections” can be a websocket connection. *Id.* claim 3 (“the communication component comprises **a websocket server** component and listens for **websocket connection** from the client device”); *Id.* ¶¶43, 61, 74, and 92; claim 8. Thus, it would also have been obvious to a POSA to have established communications between the MQSD server for data transmissions around a specific topic or sporting event by utilizing a websocket server component (the MQSD) and a web socket connection with the client device. Ex.1001 ¶450. A POSA would have recognized that doing so would further the express goals of Spivey of reducing latency in such transmissions. *Id.*

g. Element 1.7

Spivey's LDS (control server) "receive[s] live data and transmit[s] the data payloads to [MQSD (socket server)]." Ex.1017 ¶55. Data payloads may include "a score, a player name, a play, and so forth"



Id. ¶¶85, 117, Figs. 9–10. Thus, a POSA would have understood that Spivey discloses a socket server receiving from the control server at least the first event information. Ex.1001 ¶451.

h. Element 1.8

i. First event socket corresponding to the first event information germane to the first live sporting event

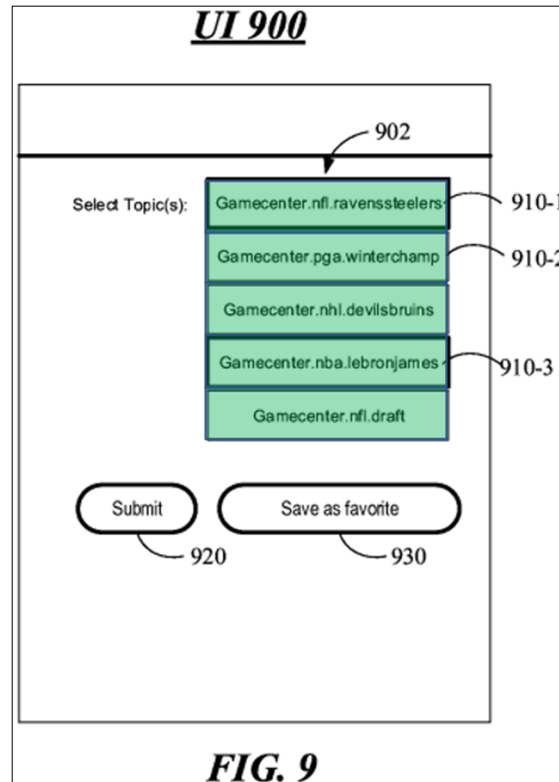
It is noted that, while Spivey often explains that a connection between the client device and the system “*may include*” a live data access server (“LDAS”) (Ex.1017 ¶38), Spivey also discloses and teaches that the message queue server device may be “accessible to clients over the network directly.” *Id.* ¶34(client and MQS may be communicatively coupled and “the system 100 may include... fewer elements in alternate topologies as desired for a given implementation”), 84. For example, “MQS communication component 308 ... may *push data* from topic queues 332–1, 332–n *to the subscribers* of the topics corresponding to those topic queues.” *Id.* ¶46. Thus, a POSA would have understood Spivey to disclose and teach a direct connection between the client devices and MQSD. A POSA would have used this implementation because reducing the amount of server transfers would reduce latency, which is Spivey’s main purpose. Furthermore, implementing the functionality of multiple servers as a single server was an extremely well-known alternative to a POSA. Ex.1001 ¶¶452–453.

A first event socket corresponding to the first event information germane to the first live sporting event is simply a socket where information about one sporting event is provided to VCDs. A POSA would have understood a “socket” to be a

uniquely addressable communication endpoint on a server where first event information is provided to a VCD. Ex.1003 20:55–21:4, 21:63–22:16; Ex.1029 14–20; Ex1030; Ex.1001 ¶¶454, 52–53. Spivey’s **topic queues** are “where data payloads are placed” that, for example, “relate to a specific game between two NFL teams.” Ex.1017 ¶¶51–52. The data payloads, “[w]hen the live event is a sporting event,” include, “for example, a score, a play, ..., and so forth.” *Id.* ¶48. Topic queues have a corresponding “topic label” which is used “to identify a topic queue.” *Id.* ¶49. “A topic label may be, for example” a “text string such as ‘gamecenter.nfl.ravenssteelers.’” *Id.* ¶66.

Client devices use these topic label text strings to connect and subscribe to MQSD’s topic queues. First, client devices “request a live data connection” with the MQSD by “select[ing] or enter[ing] the address or connection information to a particular” MQSD. *Id.* ¶¶81–82.

This “address may resemble a uniform resource locator (URL).” *Id.* ¶82. Second, as shown in Figure 9, client devices can use the topic labels to select “the topics available from [MQSD] in a topic selection list” and “subscribe to” those topic queues:



Id. ¶¶83–84, Fig. 9(annotated). Third, “the data payload” may be pushed to the subscribed connected devices that requested that topic. *Id.* ¶96. Thus, a POSA would have understood Spivey’s topic queues disclose first event sockets corresponding to the first event information germane to the first live sporting event. Ex.1001 ¶455.

Alternatively, it would also have been obvious to a POSA to have established communications between the MQSD server for data transmissions around a specific topic or sporting event by utilizing a websocket server component (the MQSD) and a web socket connection with the client device for the same reasons explained above and in §VIII.A.2.f.; Ex.1001 ¶456.

ii. Transmitting the first event information to at least the first VCD of the first plurality of VCDs

Ellis discloses a first VCD of the first plurality of VCDs. §§VIII.A.2.a., d. Both Ellis and Spivey further disclose transmitting the first event information to the first/second VCD of the first plurality of VCDs. For example, in Spivey, the MQSD “may push data from topic queues ... to the subscribers of the topics corresponding to those topic queues.” Ex.1017 ¶46; *see* §VIII.A.2.h. In Ellis, SCS may be configured “to provide real-time sports scores and game summary information, ... to user equipment 102.” Ex.1016 ¶41; Ex.1001 ¶457.

iii. Transmitting via a first event information Internet communication channel between at least one first event socket of the at least one socket server and the first VCD of the first plurality of VCDs

Ellis in view of Spivey discloses providing first event information over a separate internet communication channel. Spivey’s entire system is based on providing “live event data in real time.” Ex.1017, Abstract. The MQSD can be an “internet server.” *See id.* ¶35. Spivey communicates with client devices over “any ... combination of wired and/or wireless networks including ... the [i]nternet.” *Id.* ¶179. For example, client devices “may request and participate in a websocket connection,” which “typically comprises a two-way connection between two entities.” *Id.* ¶¶43, 61. Moreover, consistent with Spivey, Ellis discloses providing event information over a separate communication path 152 than the video signal 150.

Ex.1017 ¶¶39–41, Fig. 1. Thus, a POSA would have understood Ellis in view of Spivey to disclose transmitting first event information via a separate internet connection. Ex.1001 ¶458.

Further, this information is provided via a connection between a topic queue and the first VCD. Users can access the MQSD using a URL. *See* Ex.1017 ¶¶81–82. Then, the user can access the specific topic queue by using a text string, *e.g.*, a topic label, associated with a topic queue. *Id.* ¶66, Fig. 9 (“Gamecenter.nfl.ravenssteelers.”). Thus, a POSA would have understood that Ellis in view of Spivey discloses distributing event information from a socket to a user device over an internet connection. Ex.1001 ¶459. A POSA would have understood that the internet connection is established using a URL and text string identifying a uniquely addressable communication endpoint on the MQSD, where first event information is transmitted to a VCD. *Id.*

B. Ground 2: Claim 10 is Obvious Over Ellis in View of Spivey and Kellicker

1. Motivation to Combine

Kellicker teaches that one of the objectives of the arrangement described therein is to “[coordinate] data analytics to improve performance of streaming servers, client devices, and analytics servers.” Ex.1049 ¶4. The arrangement disclosed by Kellicker is illustrated below.

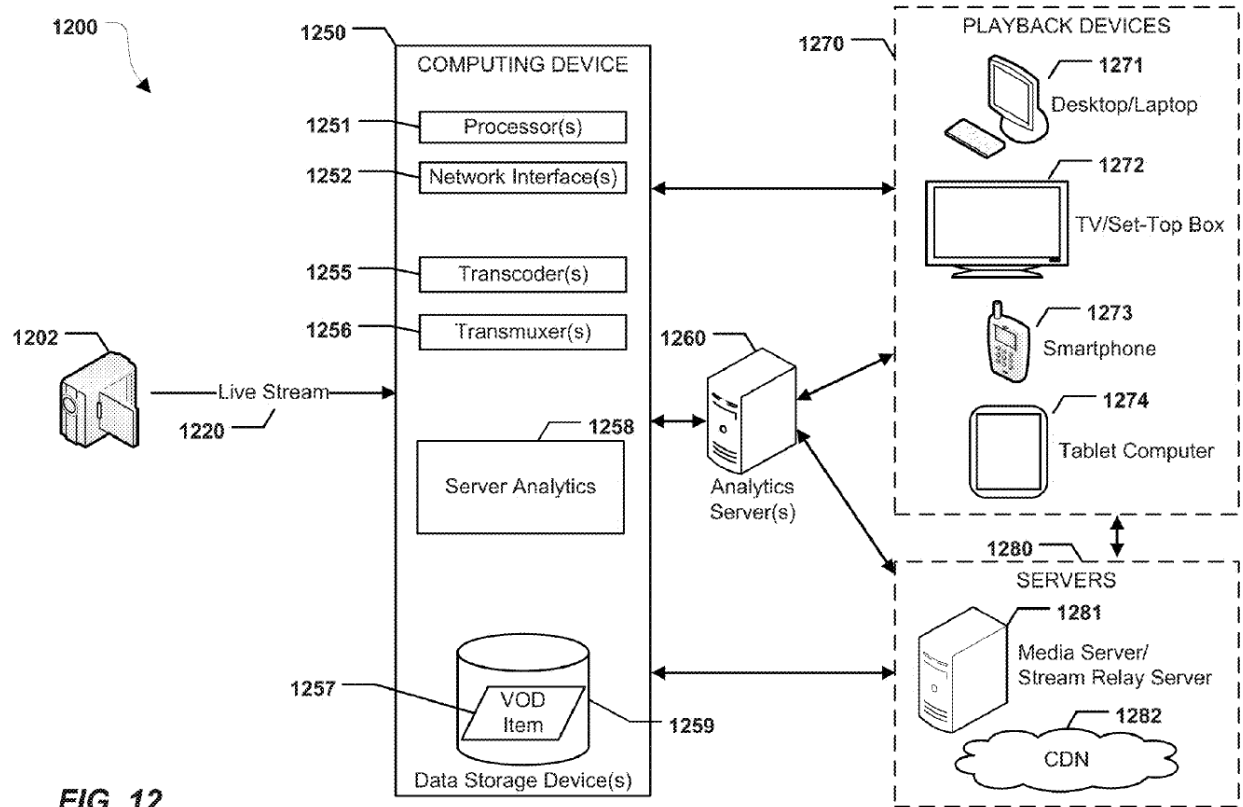


FIG. 12

Id. Fig. 12.

As evident from the above, a POSA would have understood that the “computing device” 1250 of Kellicker is analogous to the “media sources” of ‘687, as well as the content sources 130 of Ellis. Ex.1001 ¶¶115–17. The computing device 1250 of Kellicker is a server that receives incoming live media stream 1220 from a capture source (*e.g.*, broadcasting client) 1202, processes the incoming media, and serves it to playback devices 1270 either directly, or through other servers 1260, 1280. The server 1250 includes a processor, a memory and a network interface. Ex.1049 ¶¶20, 98–100; Fig. 12. Kellicker teaches that server 1250 is provided with the ability to receive a live media stream 1220, and stream that media to user devices.

Id. ¶102; Fig.12. Kellicker teaches that server 1250 supports a number of real-time data communication protocols, including HLS, RTMP and/or WebRTC. *Id.* ¶106. Server 1250 can be implemented as a single device/server or as a plurality of devices/servers. *Id.* ¶98(“the computing device 1250 may correspond to the streaming server 120”), and ¶19(“the streaming server 120 . . . may be distributed across a plurality of devices”); Ex.1001 ¶¶118–19.

Ellis, Spivey and Kellicker all address media content delivery in real-time events and describe methods and/or systems for streaming video and/or event data to one or more users. The references solve the same or similar problems, namely, providing video and/or event data to multiple users, and provide the ability to support the processing and transmission of the broadcast media utilizing a number of different well-known transmission protocols. *Id.* ¶120. A POSA would have successfully combined the teachings of Ellis, Spivey and Kellicker because such modifications involved only the substitution or addition of known elements in a predictable manner to achieve a known or desired result. *Id.*

2. Dependent Claim 10

a. Elements 10.0–10.2

Kellicker discloses a server (120/1250) that receives incoming live media streams, processes the incoming media and serves it to playback devices, including receipt of a live incoming media stream (1220), and real-time streaming of the

media. §VIII.B.1; Ex.1049 ¶¶101, 106. Kellicker discloses this server can be implemented through a plurality of such servers. *Id.* ¶¶98, 19(“may be distributed across a plurality of devices”). Thus, Kellicker discloses a plurality of media sources in the form of real-time servers. Ex.1001 ¶¶461–62.

Further, Kellicker teaches that the server(s) may support various streaming protocols, including RTMP, and WebRTC. Ex.1049 ¶106(“computing device 1250 may also support real time messaging protocol (*RTMP*) (and variants thereof)... web real-time communication (*WebRTC*).” Thus, Kellicker discloses media sources in the form of a real-time a RTMP media server and/or a real-time WebRTC server. Ex.1001 ¶463.

Kellicker teaches such server(s) “enable the computing device 1250 to process data in accordance with multiple coding technologies and protocols.” Ex.1049 ¶104. Thus, it would have been obvious to a POSA to have implemented the content source(s) 130 of Ellis in the form of either a real-time a RTMP media server or a real-time WebRTC server. A POSA would have been motivated to do so in order to provide the content source of Ellis with the processing and transmission ability to conform with these well-known and widely utilized protocols. *Id.* ¶¶103–106; Ex.1001 ¶464.

Claim 10 only requires either a RTMP server *or* a WebRTC server. Nevertheless, modifying Ellis to implement the content source(s) 130 therein in the

form of multiple servers, including a RTMP server *and* a WebRTC server would have also been obvious. A POSA would have recognized that doing so provides the system with even more flexibility and adaptability to handle different types of incoming media and comply with multiple types of protocols. Ex.1001 ¶465.

b. Element 10.3

Kellicker teaches at least one RTMP media server and at least one WebRTC media server for receiving a livestream in a corresponding protocol and transforming that livestream into another protocol for transmission to user devices. Ex.1049 ¶102–03, 106.

Kellicker teaches that the “network interface(s) 1252 [of computing device 1250] may further be configured to transmit data to the one or more other servers 1280... a media/relay server (1281), a server of a content distribution network (1282)” (*e.g.*, a regional or edge server), etc. *Id.* FIG. 12. Kellicker explains that the data transmitted to servers 1280 includes “media streams (*e.g.*, streams including media content, such as audio, video, text, still images, or a combination thereof).” *Id.* Accordingly, Kellicker teaches that the at least one RTMP media server and/or at least one WebRTC media server are communicatively coupled to the one or more other servers 1280. *Id.*; Ex.1001 ¶¶466–67.

Kellicker further discloses that the media streams received by the servers 1280 “may be associated with the same encoding format and transmission protocol or may

be associated with different encoding formats and transmission protocols.” Ex.1049 ¶102; FIG. 12. Kellicker specifically teaches providing the real-time streaming server(s) with the HLS streaming protocol ability. *Id.* at ¶106. It would have been obvious to a POSA to implement this capability in the content source(s) of Ellis in the form of the one or more other servers 1280, such as media/stream relay server 1281, for Ellis’s content source(s). A POSA would have been motivated to provide HTTP live streaming (HLS) media streams because HLS provided a reliable, cost-effective means of delivering continuous and long-form video over the Internet. Ex.1001 ¶468.

As explained in Kellicker “the other servers 1280 include a media server/stream relay server 1281 and a server of a content distribution network (CDN) 1282.” Ex.1049 at ¶102. A POSA would have readily appreciated that a CDN is a caching architecture. Ex.1001 ¶469. Thus, the HLS compatible CDN of Kellicker is a caching server architecture. Furthermore, it would have been obvious to a POSA to have provided the other server(s) 1280 architecture with caching capability in order to improve responsiveness to requests from client devices to retrieve the streamed media from the other server(s) 1280, consistent with the fundamental purpose and operation of a CDN.

c. Element 10.4

Kellicker teaches communicating served content between the HLS enabled other server(s) 1280 architecture and client devices 1270:

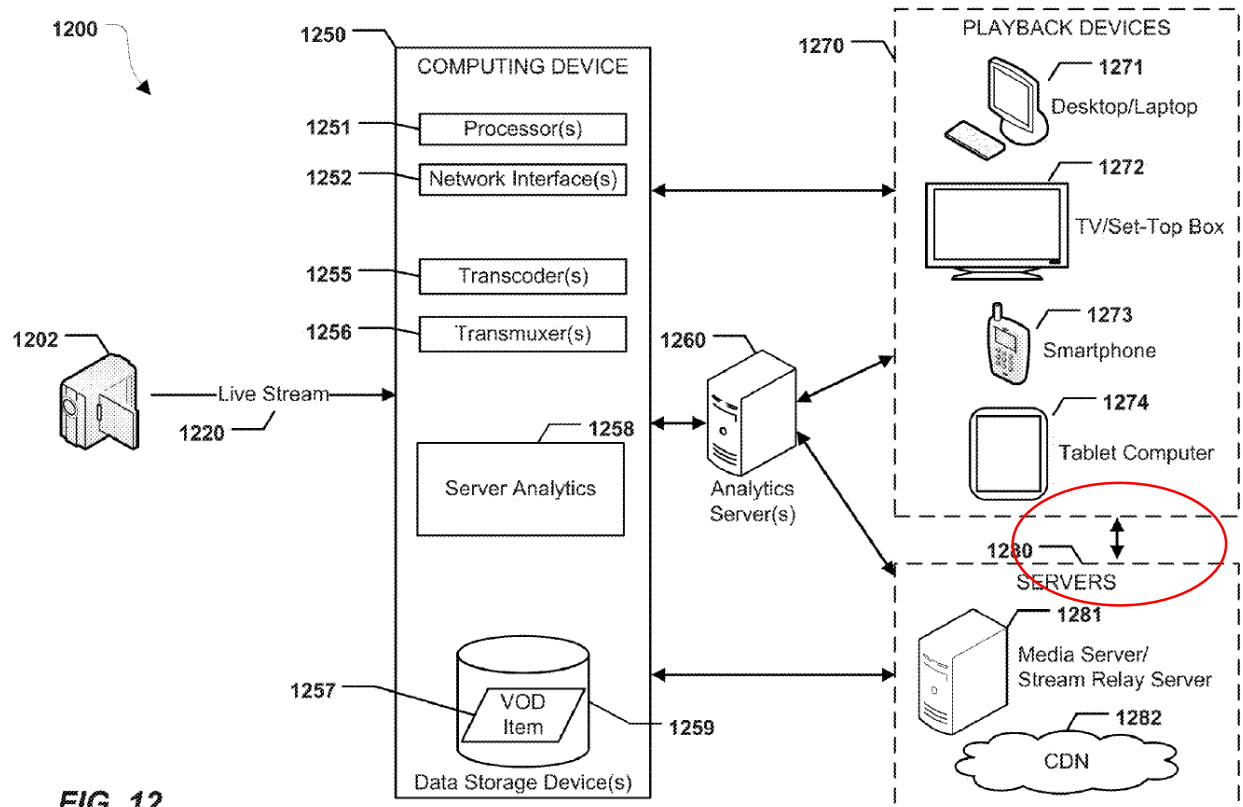


FIG. 12

Ex.1049 Fig. 12(annotated), ¶¶19, 98, 111, 102.

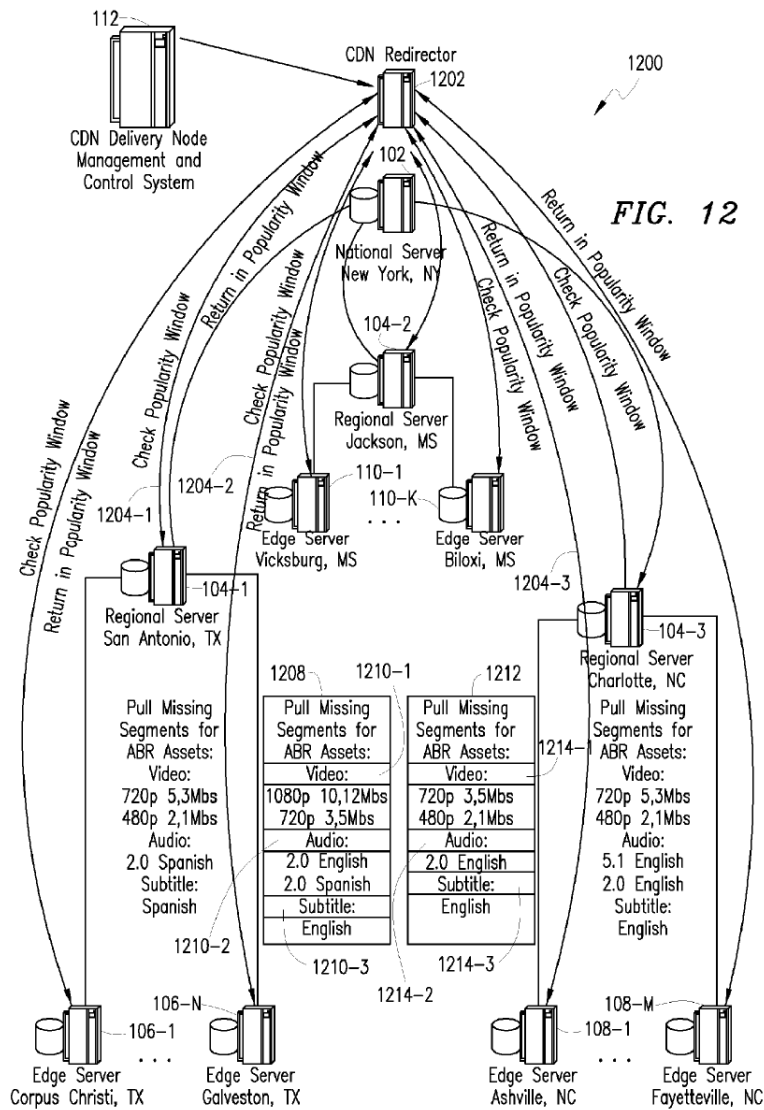
Likewise, Ellis teaches establishing a similar Internet video communication channel 160 between its contents source 130 and user equipment 102. See §VIII.A.2.d. Thus, Ellis in view of Spivey and Kellicker. Ex.1001 ¶¶470–72.

C. Ground 3: Claim 11 and 15–16 are Obvious Over Ellis in View of Spivey, Kellicker and Phillips

1. Motivation to Combine

Kellicker teaches an HLS caching server architecture (*e.g.*, 1280/1282) communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server architecture (*e.g.*, 1250). §VIII.B.2.b. Kellicker teaches this “server” may include “a content distribution network (CDN) 1282.” Ex.1049 ¶102. While Kellicker does not expressly describe the CDN architecture, conventional prior art CDN architecture routinely included both “parent” and “children” (edge or point of presence (PoP)) servers. Ex.1001 ¶¶121–23.

An exemplary CDN architecture that may be employed to distribute HLS media streams is described by Phillips. Ex.1050 ¶31 (“content may be delivered via CDN 100 using . . . HTTP Live Streaming or *HLS*”), FIG. 1. Phillips teaches one of the objectives of the CDN architecture described therein is to “[optimize] defragmentation of content in a content delivery network (CDN).” *Id.* Abstract. Phillips discloses a hierarchal server structure wherein “a central or national distribution node may in turn operate as a parent node to one or more child regional distribution nodes.” *Id.* ¶31. Phillips teaches this architecture provides “for high–performance streaming . . . to subscribers using one or more Internet–based infrastructures.” *Id.* ¶30. Ex.1001 ¶¶124–25. The CDN architecture disclosed and taught by Phillips is depicted, for example, in Fig, 12:



Ellis, Spivey, Kellicker and Phillips all involve media content delivery systems including real-time streaming of live or real-time media. Each describe methods and/or systems for streaming video and/or event data to one or more users. The references solve the same or similar problems, namely, providing video and/or event data to multiple users, and provide the ability to support the processing and transmission of the broadcast media utilizing several different well-known

transmission protocols. Ex.1001 ¶126. A POSA would have successfully combined Ellis, Spivey Kellicker and Phillips, as such modifications involved only the substitution or addition of known elements in a predictable manner to achieve a known or desired result. *Id.*

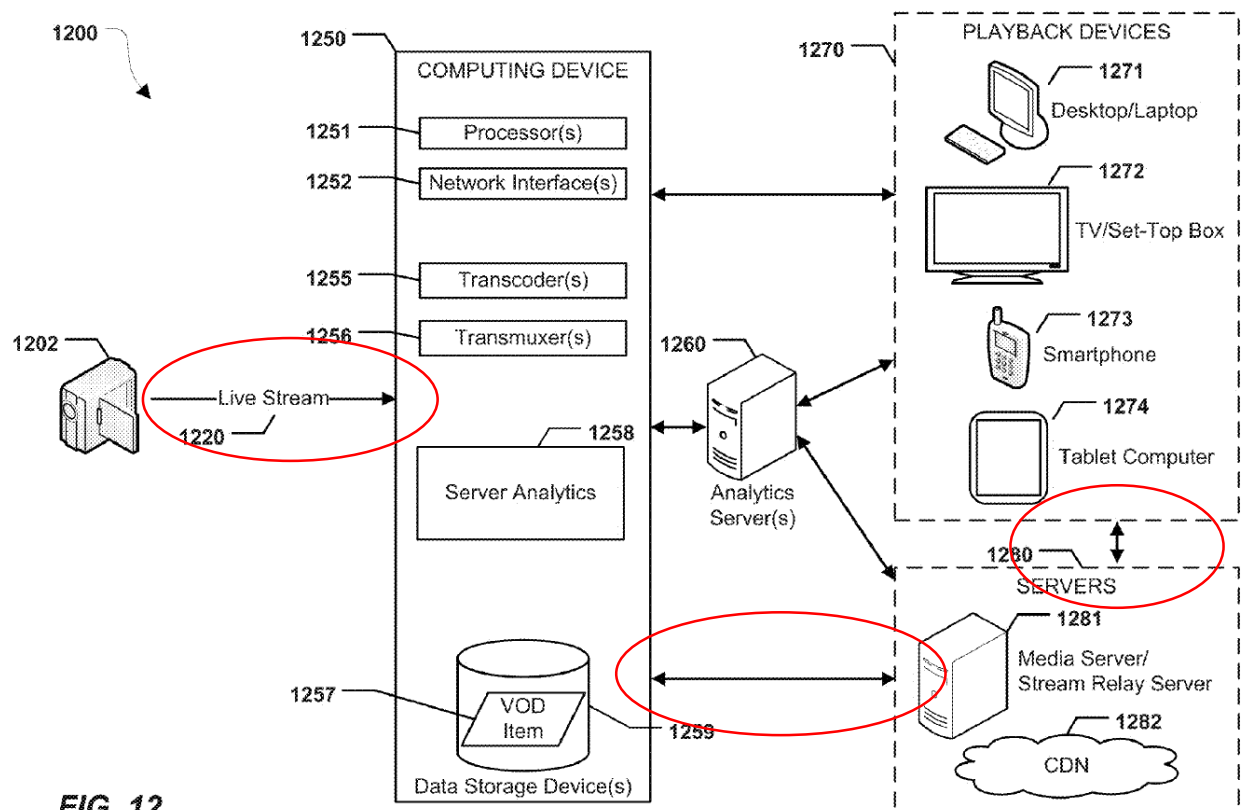
2. Dependent Claim 11

a. Elements 11.0–11.2

A POSA would have found it obvious to provide the HLS caching server CDN, taught by Kellicker, in the form of a hierarchal server structure wherein “a central or national distribution node may in turn operate as a parent node to one or more child regional distribution nodes.” Phillips ¶31. Consistent with Phillips, a POSA would have been motivated to do so in order to provide “for high–performance streaming... to subscribers using one or more Internet–based infrastructures.” *Id.* ¶30. For example, Phillips teaches this hierarchal server structure provides the ability to optimize defragmentation of content in the CDN. *Id.* ¶57. For instance, “[w]hen edge servers do not hold a copy of a requested segment file, they may be required to retrieve or pull the requested segment file from a parent node.” *Id.* ¶58, 59. A POSA would have been further motivated to provide the claimed parent/child HLS caching server CDN architecture to achieve greater CDN efficiency in the delivery of content. *Id.* ¶67(“the embodiments of the present

disclosure... allow CDNs to function more efficiently by making available what is needed where it is needed”). Ex.1001 ¶¶474–475

As explained above, Kellicker, as modified by Phillips, establishes a communication path from a first broadcaster's livestream (1220) to a first server architecture (1250), which may be implemented as a plurality of media sources including at least one RTMP media server and/or at least one WebRTC media server, to a CDN in the form of parent/child HLS caching server CDN architecture (1282), and from the CDN to a client/user device (1270):



Ex.1049, Fig.12(annotated).

Specifically, Kellicker discloses that media server(s) 1250 is configured to transmit data to “a server of a content distribution network (CDN) 1282.” *Id.* ¶102. Further, Kellicker discloses “[i]n an embodiment where the analytics server(s) 1260 is collocated with a streaming server (*e.g.*, the computing device 1250), first HTTP port(s) may be used to communicate a master rendition list and media chunks . . .”. *Id.* ¶110; Ex.1001 ¶¶476–77.

Similarly, Phillips teaches that content requested for delivery by the CDN is received in the form of a “manifest” that is made available to a “content delivery node” of the CDN server architecture. To the extent the content delivery node does not possess the requested files, they can be retrieved from another node in the CDN architecture, such as an edge server node (*i.e.*, “child” server). Ex.1050, Abstract, ¶31, Fig. 1. To the extent that the nature/content of the files being handled by a system is considered as further limiting the claimed system, a POSA would have understood that a “manifest” or a “master rendition list” constitutes an “HLS file suite.” Ex.1001 ¶478. Phillips’s CDN architecture discloses mother servers (*e.g.*, national server, regional servers), which are mother servers to children servers (*e.g.*, regional servers, edge servers). A POSA would have readily appreciated that the mother–child relationship is defined in the hierarchy itself. The regional child servers are in–turn coupled to their children (*e.g.*, edge servers). These edge servers form an HLS content delivery network (HLS CDN), such that, as discussed below,

“the system establishes the first video Internet communication channel... between the first viewer client device and the HLS CDN.” Ex.1050 ¶31, FIGS. 1, 12; Ex.1001 ¶478.

Furthermore, the “HLS file suite” represents nothing more than conventional HLS file transfer protocol, as admitted by the Patent Owner (PO). Ex.1003, 38:1–9. A POSA would have understood that the transfer of media according to the HLS standard, as taught by both Phillips and Kellicker includes the use of exchanged HLS file suites. Ex.1001 ¶479.

Thus, the arrangement of Kellicker as modified by Phillips includes a mother server of a CDN that is “communicatively coupled... to receive... a first HLS file suite representing the first broadcaster's live stream of digital content.” A node which receives the output from the RTMP and/or WebRTC media sources and serves it to another server in the CDN would operate as an HLS mother server. Ex.1001 ¶480.

Alternatively, and to the extent that the nature or content of the data could be considered as somehow limiting the system that handles it, this aspect of the claimed invention would have been obvious to a POSA as the appropriate and defined mechanism for accomplishing successful file transfer in accordance with the HLS protocol. Ex.1001 ¶¶481–82.

b. Element 11.3

Ellis, as modified by Spivey, establishes the first video Internet communication channel to provide the first copy of digital content to the first viewer client device. §VIII.A.2.d. Ellis, as modified by Spivey, Kellicker and Phillips, establishes that the first video Internet communication channels provided between the first viewer client device and the HLS CDN. §§VIII.B.2.b.–c; VIII.C.2.a; Ex.1001 ¶¶483–84.

c. Element 11.4

Ellis, as modified by Spivey, Kellicker and Phillips, establishes that the first copy of the first broadcaster's live stream of digital content is based on the (first) HLS file suite. §VIII.C.2.a; Ex.1001 ¶485.

3. Dependent Claim 15

a. Element 15.0

The recitation in claim 15 of “*the HLS mother server requests, from the at least one of the at least one RTMP media server and the at least one WebRTC media server, only the first HLS file suite representing the first live stream of digital content and no other HLS file suite*” amounts to nothing more than a recitation of intended use of an HLS mother server/child server CDN architecture and does not serve to further limit the claimed system. The HLS CDN caching server architecture disclosed by Phillips is the same as the claimed HLS mother/child server

architecture, and therefore is perfectly capable of executing the claimed intended use, as explained below. Ex.1001 ¶¶486–87.

Furthermore, a POSA would have understood this element recites nothing more than conventional functionality of conventional caching CDN network. Ex.1001 ¶488.

Phillips discloses that the national server 102 of its CDN network is one example of a *mother server*. Ex.1050 ¶¶31, 32, 56, Fig. 12. As a POSA would have understood, this means that the national/mother server of Phillips can act as the primary server where the original/complete version of the served content is stored, acting as a source for content created for distribution by the Content Delivery Network (CDN) which distributes cached copies of that content from geographically distributed edge servers closer to users for faster access. In other words, it can act as a backend server that holds the master copies of all created content that is served through the CDN network. Ex.1001 ¶¶489–90.

Further, it would have been obvious to a POSA, in view of Phillips, to have used the mother server to request “only” the first HLS file suite associated with a broadcaster's live stream of digital content and “no other” HLS file suite, consistent with the conventional and well understood functionality of a mother or caching server. This content could then be served directly to child servers within the CDN without making another request to the content creator. Ex.1001 ¶¶491–93. This

implementation ensures the integrity of the data being served throughout the network by maintaining a master copy at a centralized location which is then replicated and served by the network, reduces load on the content creator server structure, and facilitates the defragmentation objective taught by Phillips. *Id.*

4. Dependent Claim 16

a. Element 16.0

Initially, it is noted that the characterization of a “*persistent connection*” between the mother server and the media sources as being “*during at least a portion of a duration of the first live stream*” renders the term “persistent” meaningless. This is particularly true given that the claim does not offer any limitation on the “*duration*” of the stream, which could, therefore, be on the order of a millisecond. Thus, claim 16 is also obvious on at least the same basis as claim 15, because *any* connection between the HLS mother server and the media sources would be “persistent” over some “duration” of time. Ex.1001 ¶¶494–95.

Further, it would have been abundantly clear to a POSA that the very nature of a live stream transmission involves a “persistent” connection between the source of the media and the server distributing it to a user/client, as admitted by ‘687:

A *live stream* of digital content may have a variety of data
... and transmission protocols, *including persistent/continuous
streaming transmission protocols* (e.g., real time streaming

protocol ‘RTSP;’ real time messaging protocol ‘**RTMP**;’ web
realtime communication ‘**WebRTC**’), as well as segmented
and/or adaptive bitrate (ABR) protocols (*e.g.*, Apple's HTTP 45
Live Streaming ‘**HLS**;’ . . .

Ex.1003, 8:45–55; Ex.1001 ¶496.

The ‘687 alleges that “HLS connections are, by definition, not persistent connections *between the HLS media server and the viewer client device*, since requests for and delivery of HLS content uses only standard HTTP transactions.” Ex.1003, 33:61–65. However, claim 16 refers to the connection between the media source and the HLS CDN mother server—*not* a connection between a server and a client device. *Id*; Ex.1001 ¶497.

Thus, a POSA would have readily understood the connection between the RTMP/WebRTC media source(s) and the HLS CDN mother server, as taught by the arrangement of Ellis in View of Spivey, Kellicker and Phillips, defines a “persistent connection.” This is especially true considering that the “*persistent connection*” only need be persistent for “*a portion of a duration of the first broadcaster's live stream of digital content.*” Ex.1001 ¶498.

Alternatively, given the nature of the connection between the RTMP media server, the WebRTC media server, and the HLS mother server, it would have been obvious to a POSA to utilize a persistent connection because a non–persistent

connection, like an HTTP (request, response, close) connection, would be less efficient and have more overhead. Ex.1001 ¶¶499–500.

Claim 16 also requires that the persistent connection is maintained during a portion of a duration of the first broadcaster's live stream to receive respective elements of the first HLS file suite. In other words, the claim requires that some portion of the respective elements of the first HLS file suite be received by the HLS mother server during the live stream. This is consistent with live streaming itself. For example, receiving all elements of the first HLS file suite were at the HLS mother server after the live stream, would merely be playing a recorded video (VOD). Because Kellicker supports live streaming (as well as VOD) the HLS mother server (*e.g.*, server 1280) would maintain connection to the at least one RTMP media server and/or at least one WebRTC media server, during at least a portion of a duration of the first broadcaster's live stream of digital content, to receive respective elements of the first HLS file suite. *See e.g.*, Ex.1049 ¶¶24,28–30. Moreover, this condition will always be satisfied by a persistent connection that persists for the entirety of the broadcaster's live stream. Ex.1001 ¶501–02.

Thus, for at least the reasons explained above, it would have also been obvious to a POSA implement a persistent connection between the RTMP/WebRTC media source(s) and the HLS CDN mother server. Ex.1001 ¶494.

D. Ground 4: Claims 12–14 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips and Stoica

1. Motivation to Combine

Stoica is directed to methods and arrangements for using networks, such as the Internet, to access content including video files and live streaming via client devices. Ex.1051, 1:15–17. According to certain embodiments, Stoica discloses a video serving method and arrangement using the HTTP chunking delivery protocol. *Id.* 5:55–60. Such an arrangement is depicted in Fig. 8:

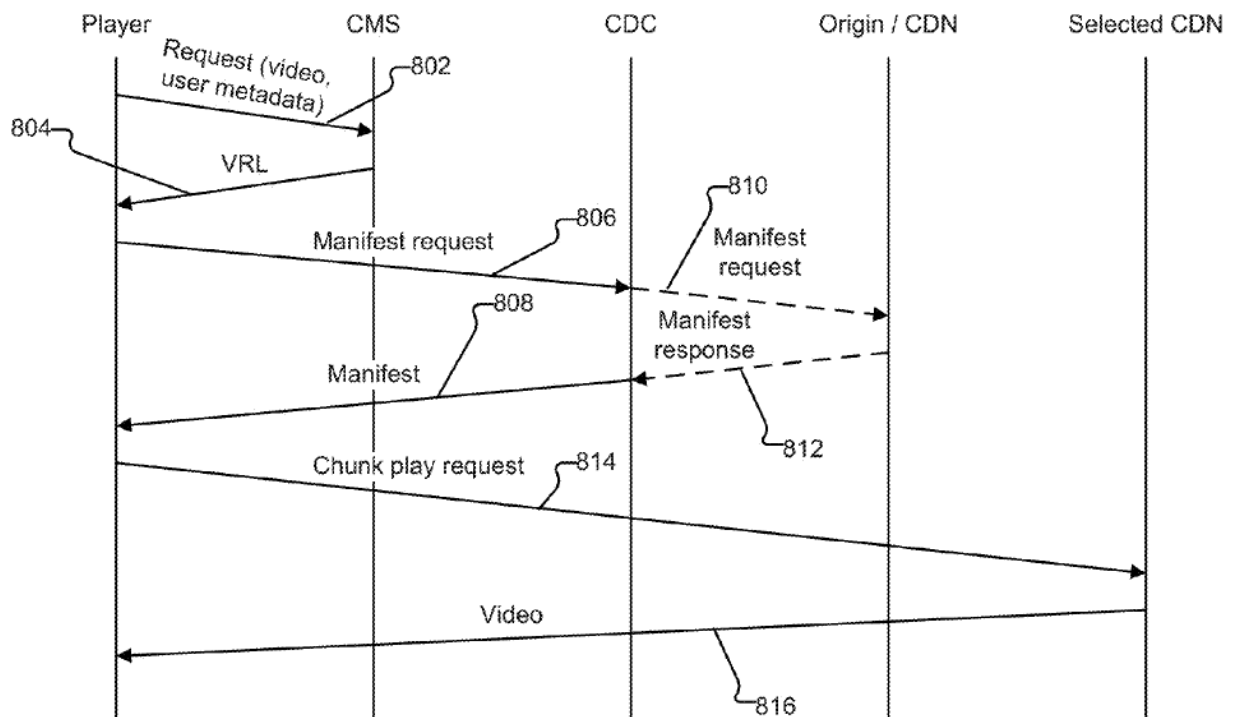


FIG. 8

Both the Content Management System (CMS) and Content Distribution Coordinator (CDC) act as a web server that facilitate the streaming transaction as an

intermediary between the client (player) and the content generated by a video file provider through a content distribution network (CDN). *Id.* 5:55–6:54. For instance, “the CMS provides metadata associated with a video stream, including the information required to deliver the content.” *Id.* 5:62–64; Ex.1001 ¶¶127–29.

Stoica also discloses the use of a gateway 1122 and dispatcher server 1114, each in the form of a web server configured to handle web service API calls in various formats. Stoica further discloses that the gateway and dispatcher server are coupled to a database for storing information regarding the API request, data access, etc. Ex.1051, 9:63–10:5;11:17–32; Ex.1001 ¶130. The arrangement is illustrated Fig. 11:

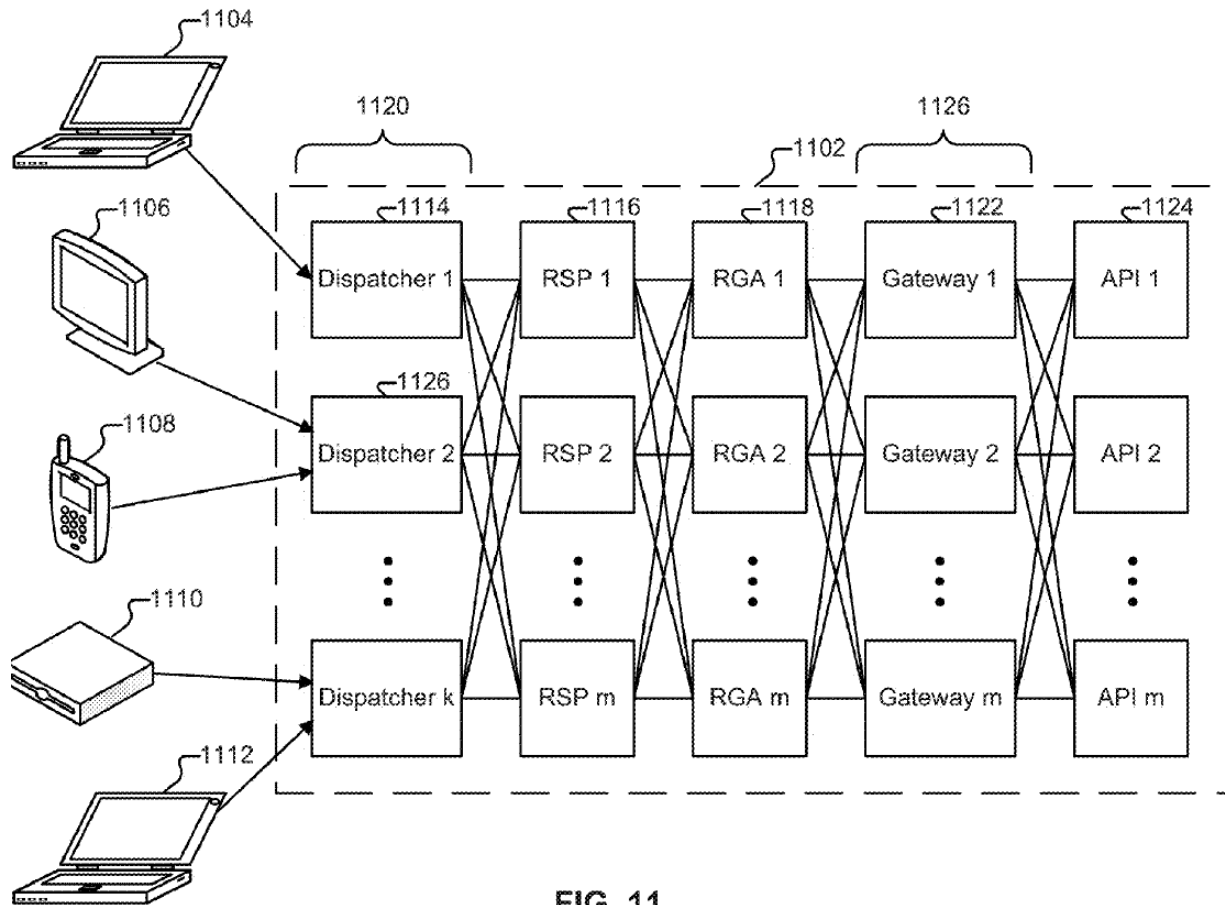


FIG. 11

For example, it would have been obvious to a POSA to implement such a web server architecture in order to facilitate communications and delivery of content between client and the content source. Ex.1001 ¶131.

Additionally, Stoica teaches that the implementation of its methods and arrangements addresses demand for efficient distribution of online media content, including video, with high quality. Ex.1051, 1:18–20, 9: 63–10:2, Fig. 11; Ex.1001 ¶131.

Ellis, Spivey, Kellicker, Phillips and Stoica all involve media content delivery systems including real-time streaming of live or real-time media. Each describe

methods and/or systems for streaming video and/or event data to one or more users. The references solve the same or similar problems, namely, providing video and/or event data to multiple users, and provide the ability to support the processing and transmission of the broadcast media utilizing well-known transmission protocols. Ex.1001 ¶132. A POSA would have successfully combined Ellis, Spivey, Kellicker, Phillips and Stoica because such modifications involve only the substitution or addition of known elements in a predictable manner to achieve a known or desired result. *Id.*

2. Dependent Claim 12

a. Elements 12.0–12.1

Stoica teaches at least one webserver.

First, Stoica teaches a webserver that provides an API⁴:

⁴ Petitioner notes that the recitation of “*to provide an applications programming interface (API)*” amounts to nothing more than a statement of intended use. The claimed “API” does not specify any functionality, nor is the claimed “API” further mentioned in claim 12 or claims dependent thereon. Accordingly, the claimed webserver merely needs to provide *any* API accessible by *any* other communicatively coupled device. Ex.1001 ¶505.

In some embodiments gateway 1122 is implemented using an off-the shelf web server (such as Apache) with customized code to handle the various web service API calls. The handlers return data for a given web API call in various data formats including XML, JSON, SOAP, and HTML, as applicable. The handlers serve as middleware for querying and interactively controlling the RSPs and RGAs.

Ex.1051 11:17–27.

Second, Stoica also teaches a webserver in the form of a “dispatcher server” (1114) used to gather status information from streaming clients:

A dispatcher server (1114) includes... [a] first module implement[ing] a communication interface for receiving status information from clients. In some embodiments the module is implemented using an off-the-shelf web server, and allows clients to connect over the HTTP protocol (and also allows clients to securely communicate via SSL.

Id. 9: 63–10:2, Fig. 11.

Gateway 1122 and dispatcher server 1114 of Stoica are each communicatively coupled with both client devices and the requested content. *Id.* 9: 63–10:2; 11:17–32; Fig. 11. As further disclosed by Stoica, such content is delivered through a CDN.

Id. 2:49–53. The requested content is provided to the CDN according to the system architecture taught by previous combinations of cited references as described herein, including first and second pluralities of broadcast clients⁵ and viewer clients. §§VIII.A.2.a; VIII.C.2.a.–b. Thus, gateway 1122 and/or dispatch server 1114 of Stoica satisfies these elements. Ex.1001 ¶¶504, 506–08.

Third, Stoica teaches that the CMS is a webserver that provides an applications programming interface (API) because it interfaces between a client's “web browser” and the requested content through a CDN:

operator of a ... client device 402, wishes to watch a video using video player application 404... the player application is provided by a web browser application... client device 402 connects to Content Management System (CMS) 406 (*e.g.*, via one or more networks 110) and requests the content.

Ex.1051, 4:11–17, Fig. 8, 5:62–64(CMS provides metadata associated with a video stream, including the information required to deliver the content.”).

In this regard, Stoica teaches the CMS/CDC can provide an augmented URI (*i.e.*, VRL) specifying a manifest file to a client device:

⁵ It is noted that there is no antecedent basis for “the first broadcaster client device” or for “the second broadcaster client device” as recited in elements 12.1 and 12.2.

Specifically, instead of including a traditional URL to the manifest file, the CMS returns a URL that points the client to the CDC. This URL is also referred to herein as a “Virtual Resource Locator” or VRL... [w]hen the CDC is accessed via a VRL (806), the CDC dynamically generates and sends to the client (808) a manifest file

Id. 6: 26–34, FIG. 8(illustrating video request, VRL response, manifest request, and manifest response); 6:62–67(“CMS provides... the URL of a manifest file. In particular, the manifest file contains the URL names of the chunks encoding the content, and can also include the URL/names of other manifest files.”). Ex.1001 ¶¶509–10.

The CMS/CDC are communicatively coupled with both client devices and the requested content. Ex.1051, Fig. 8. As further disclosed by Stoica, such content is delivered through a CDN. *Id.* 2:49–53. The requested content is provided to the CDN according to the system architecture taught by previous combinations of cited references as described herein, including first and second pluralities of broadcast clients and viewer clients. §VIII.A.2.a., d. Thus, the CMS/CDC of Stoica satisfies these elements. Ex.1001 ¶511.

b. Element 12.2

Stoica teaches that the at least one of the webserver is communicatively coupled to a memory system including a database.

Stoica teaches that the dispatcher server 1114 is communicatively coupled to a database storing stream information:

A dispatcher sends all heartbeats belonging to the same session to the same RSP component... The RSP component reads network data one record at a time and parses the data into a local data representation. The data received by the RSP component can be stored as in-memory hash tables of records allowing fast execution, and very high throughputs... [i]n other embodiments, optimized in-memory databases are used.

Ex.1051, 10:8–24.

To the extent the types of labels/content being stored or used is considered as further limiting the claimed system, Stoica explains that the database stores a stream identifier (StreamID) (*e.g.*, a “show” identifier), which is received for the first broadcaster's live stream of digital content:

Once an RSP component parses the data records, it performs two main tasks. First, it performs data filtering. A filter is a logical expression and is installed at each RSP component

instance. As one example, the following filter would identify 35 viewers located in San Francisco, connected to ISP SP1, streaming from CDN A, one of two particular shows: (city=“ San Francisco” AND I SP=“ SP1 “ AND CDN=“CDN A” AND ((show=“NewsAt10”) OR (show=“SundayMagazine”)).

Id. 10:31–49.

It is noted that, by storing a stream identifier (StreamID) for the first broadcaster's live stream of digital content the database stores first stream information for the first broadcaster's live stream of digital content including the first live sporting event video–based commentary⁶ from the first broadcaster client device because such data by definition includes the stream identifier (StreamID). *Id.*; Ex.1001 ¶¶512–15; §VIII.A.2.a.

Similarly, Stoica teaches that gateway 1122 is coupled to a database storing stream identifier (StreamID) information:

Gateway 1122 also provides access controls and persistently stores the information regarding the API requests, data access and presentation policies, and filter descriptions. The

⁶ It is noted that there is no antecedent basis for the recited “the first live sporting event video-based commentary.”

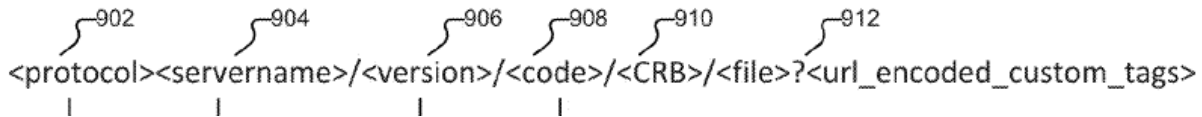
information is maintained in a persistent database, such as
mySQL or Oracle database

Ex.1051, 11:28–32.

Stoica further teaches that, through the gateway 1112 API, data is available on a per filter basis. *Id.* 11:17–19. As would have been understood by a POSA, and as evident from the disclosure cited above in connection with the discussion of the dispatcher server, stream identifier (StreamID) information is utilized for filtering, and ultimately routing content requests and retrieval. Ex.1001 ¶¶516–17. Thus, a POSA would have understood that Stoica also teaches that gateway 1122 is coupled to a database storing stream identifier (StreamID) information. The StreamID is a unique identifier for particular source of content, *i.e.*, a “show.”

c. Element 12.3

As discussed above, Stoica teaches a content management server (CMS), which may be a webserver. Stoica further explains that the CMS provides addresses (VRLs), which invoke the CDC. Ex.1051 7:4–6 (“a VRL is provided by a content provider's CMS to a client requesting content, and is used by the client to invoke the services of the CDC.”). FIG. 9(excerpted below) depicts an exemplary VRL:


The diagram shows a URL structure: <protocol><servername>/<version>/<code>/<CRB>/<file>?<url_encoded_custom_tags>. Above the URL, six curly braces with numbers point to specific parts: 902 points to <protocol>, 904 points to <servername>, 906 points to <version>, 908 points to <code>, 910 points to <CRB>, and 912 points to <url_encoded_custom_tags>.

In turn, the CDC determines the address for the HLS CDN:

To control the selection of the CDN... an indirection mechanism is used.... the CMS returns a URL that points the client to the CDC.... When the CDC is accessed via a VRL (806), the CDC dynamically generates and sends to the client (808) a manifest file based on the following information: (1) information embedded in the URL... and the CDNs or origin servers where the content is stored . . .

Id. 6:25–39; FIG. 8(illustrating transfer of video from CDN to client player).

Stoica further teaches that a number of optional parameters 912 may be embedded in the VRL, including a “show” parameter (*i.e.*, a StreamID). *Id.* 7:25–26, 8:1–4(“The following is a list of example optional parameters that can be embedded in portion 912... ‘show’: ‘PopIdol’ ‘episode’: ‘S8:E5’”). This “show” parameter is the same as the StreamID stored in the memory database of the RSP component (discussed above), which is communicatively coupled to the webserver portion of the dispatcher server. Ex.1001 ¶¶518–21.

d. Element 12.4

As explained above, both the VRL and the manifest file (which includes the address of the HLS CDN) are transmitted to the client, which is a conventional and necessary aspect of HLS streaming service retrieval:

CMS returns a URL that points the client to the CDC. This URL is also referred to herein as a “Virtual Resource Locator” or VRL. . . . When the CDC is accessed via a VRL (806), the CDC dynamically generates and sends to the client (808) a manifest file based on the following information: . . . the CDNs or origin servers where the content is stored

Ex.1051, 6:26–39; Ex.1001 ¶¶522–24.

3. Dependent Claim 13

a. Elements 13.0–13.1

Ellis in view of Spivey disclose and teach media source(s) that provide a first copy of a first broadcaster's live stream of digital content of a live sporting event to a viewer client device via a first video Internet communication channel. §§VIII.A.2.a., d. Ellis in view of Spivey also discloses a socket server to transmit first score information to at least a first viewer client device via a first event information Internet communication channel between at least one first event socket and the first viewer client device. §§VIII.A.2.f., h; Ex.1017 ¶92.

To the extent the type of label(s)/content (*e.g.*, EventID) stored and used is considered as further limiting the claimed system, this aspect of the claimed invention is satisfied by Ellis, as modified by Spivey. Spivey discloses that the media content being served to a user using topic queues “where data payloads are placed” that, for example, “*relate to* a specific game between two NFL teams.” Ex.1017 ¶¶27, 51–52. These, topic queues include topics, each having a “topic label” (*i.e.*, an EventID).” *Id.* ¶¶48, 103. “A topic label may be, for example” a “text string such as ‘gamecenter.nfl.ravenssteelers.’” *Id.* ¶66; Fig. 9. this topic label (EventID) “relates to” a specific sporting event (“ravenssteelers”; StreamID). Ex.1001 ¶¶525–27.

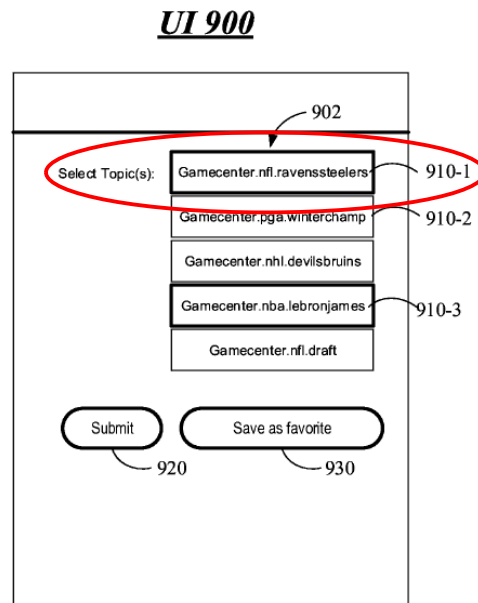
Thus, Ellis, as modified by Spivey satisfies these elements, and it would have been obvious for a POSA to store these EventIDs, in order to, for example, filter and store analytics information. Ex.1051, 9:46–10:54; Ex.1001 ¶¶528–29.

b. Elements 13.2–13.3

Spivey discloses that its MQSD and/or LDS may be/include webserver. Ex.1017 ¶¶53 (“message queue server 120 may be implemented with at least two separate devices: a message queue server device 410 and a live data server device 450.”), ¶35 (“The system 100 may include a message queue server 120. Message queue server 120 may comprise one or more electronic devices capable of receiving, processing, and sending information for the system 100. Examples of an electronic

device may include... a server, a server array or server farm, a **web server**, a network server, an Internet server”), Fig.4; Ex.1001 ¶¶530–31.

Spivey explains “UI 900 may list the topics available from a message queue server in a topic selection list 902. Topic selection list 902 may be, for example, a drop-down or pop-up menu, a list of selectable buttons, a text list having selectable buttons corresponding to the topic text, and so forth.” Ex.1017 ¶84, Fig. 9 (illustrating topic label text string “gamecenter.nfl.ravenssteelers”):



It would have been obvious to a POSA in view of these teachings of Spivey to have established a communications channel between the MQSD server for data transmissions around a specific topic or sporting event by utilizing a web socket server component (the MQSD) and a web socket connection with the client device.

§VIII.A.2.f. Further, to the extent the types of labels are considered as limiting the claimed system, a POSA would have understood that an address for the at least one first event socket of the at least one socket server is based at least in part on the EventID (*e.g.*, topic label). Ex.1017 ¶94(“Subscribing may include, for example, registering with message queue server 120 to receive any data for the requested topic, or providing an address or connection to message queue server 120 to receive any data for the requested topic.”); §§VIII.A.2.g–h. Spivey teaches that the payloads delivered to a user via the socket server on the established event communications channel further includes “live data.” Ex.1017 ¶¶55(“LDS communication component 418 may include hardware and/or software that allows live data server device 450 to receive live data and transmit the data payloads to MQS device 410”), 44, Figs. 9–10; Ex.1001 ¶¶532–33.

Stoica teaches implementation of a webserver architecture as an intermediary between a client and a content production and distribution system. §§VIII.D.2.a.–d. Stoica teaches that doing so facilitates communications and efficiency between the client and the requested content. Included in these communications is the necessary identification information to route the client's request for data to the appropriate location, thereby establishing an appropriate communications channel. Thus, it would have also been obvious to a POSA, and in view of Stoica, to transmit the first

socket address to the client to establish the first event information Internet communication channel via an intermediate web server. Ex.1001 ¶¶534–35.

4. Dependent Claim 14

a. Elements 14.0–14.1

Ellis in view of Spivey discloses transmitting information via individual communication channels between a socket and a first VCD of a first/second plurality of VCDs who are watching the same sporting event. §§VIII.A.2.a., h.

To the extent the nature/content of the information being handled by a system (e.g., “chat”) is considered as limiting that claimed system, this aspect is met by Ellis, as modified by Spivey. Ellis discloses that “sporting event” or “sports–related content” includes chat information, *i.e.*, “chat rooms,” associated with a particular event. Ex.1016 ¶¶38, 40. Ellis also discloses that “game commentary from a variety of sources may also be displayed to the user... [and] may include text... and discussion forums” and users “may also leave feedback regarding the commentary that other users in the media system may view when they access the same commentary.” *Id.* ¶132. A POSA would have understood that Ellis discloses a system that transmits and receives first chat information regarding the first live sporting event in the first broadcaster’s live stream of digital content. Ex.1001 ¶¶536–39; Ex.1016 ¶126.

Ellis also discloses providing multiple forms of content to users from multiple sources using separate communication channels. *Id.* ¶39 (“[I]n practice there may be numerous instances of content source 130 and supplemental content source 120, for clarity only one instance of each source has been shown in the example of FIG. 1.”). For example, Ellis discusses transmitting the video “broadcasts” over communication path 150 and “game summary information, ... and any other sports–related information” over communication path 152. Ex.1016 ¶41. Ellis provides that in “some embodiments, this sports–related information is provided to user equipment 102 as a plurality of real–time data feeds. Each data feed may be individually accessed or subscribed to by user equipment 102.” *Id.* ¶41.

It would have been obvious to a POSA to have established these communication channels/connections with web socket connections between the client and a web socket server. §§VIII.A.2.f.–h; Ex.1001 ¶¶540–41.

Additionally, Ellis in view of Spivey teaches a socket specifically designated for chat/system event information. For example, Spivey utilizes topic queues to organize “data payloads” that “relate to a specific game between two NFL teams.” Ex.1017 ¶¶51–52; §§VIII.A.2.f., h. The data payloads, “[w]hen the live event is a sporting event,” include, “for example, a score, a play... and so forth.” Ex.1017 ¶48. Topic queues may be “hierarchically related.” *Id.* ¶52. This means that when the system looks at the text string associated with the data payload, the system “may

place the data payload only in the topic queue exactly matching the topic label.” *Id.* ¶104; Ex.1001 ¶542. For example, if there were two topics “at the same hierarchal level” the data payloads placed in “gamecenter.nfl.ravenssteelers.statistics” would not be placed in” another topic queue “gamecenter.nfl.ravenssteelers.plays.” Thus, the statistics are separate from the chat information according to the routing methodology applied by Spivey's socket server. Ex.1017 ¶104. Thus, it would have been obvious to a POSA in view of Spivey to provide a first chat/system event information socket (“gamecenter.nfl.ravenssteelers.chat”) corresponding to the first chat information regarding the first live sporting event video–based commentary. Ex.1001 ¶543.

Thus, a POSA would have understood that Ellis, as modified by Spivey, discloses transmitting, via at least one first chat/system, event Internet communication channel between at least one first chat/system event information socket of the at least one socket server and the first VCD of the first plurality of VCDs. *Id.* ¶544.

b. Elements 14.2–14.3

Spivey teaches that event information such as statistics would have been kept separate from chat information (on a separate channel/socket). §VIII.D.4.a. Accordingly, a POSA would have understood that such a separate “chat” channel would be identified with a representative topic label in the associated hierarchy in

order to allow a subscription by a client. Because chat channels would have their own topic labels for subscribing, “the at least one webserver further determines a second socket address for the at least one first chat/system event socket of the at least one socket server” for the same reasons discussed with respect to limitation 14.1 (§VIII.D.4.a.). Ex.1001 ¶¶545–46.

In such a combined system, a POSA would have been motivated to derive EventIDs from StreamIDs in order to increase efficiency by normalizing representations consistent with Spivey. A StreamID as discussed in Stoica would have been, for example, a “show.” §§VIII.D.2.a.–b. In the context of Spivey’s examples, an exemplary “show” is a particular sporting event and assigned the nomenclature “ravenssteelers.” And topic labels (EventIDs) “related to” that sporting event (StreamID) are normalized thereto, as for example: “gamecenter.nfl.ravenssteelers.statistics,” “gamecenter.nfl.ravenssteelers.plays,” etc. Ex.1017 ¶104. In this example, the second socket address (e.g., “gamecenter.nfl.ravenssteelers.chat”) for the chat/system event socket would be based in part on the StreamID (“show” = “ravenssteelers”). Ex.1001 ¶547.

Alternatively, to the extent it is argued that such second socket address would be based on the “EventID” rather than the “StreamID,” using a StreamID rather than an EventID would be nothing more than a choice of labelling nomenclature to achieve exactly the same result—the accurate routing of specific requested content

through separate communications channels, thus representing nothing more than obvious alternative to a POSA. Ex.1001 ¶548.

Thus, it would have been obvious, in view of Spivey, for the webserver to determine a specific socket address (second address) based on a unique stream identifier (StreamID) in order to permit accurate routing of different forms of content (*e.g.*, chat) associated with a particular sporting event (StreamID) based on a selection made by the user, and to transmit the address information to the client device to establish the communications channel and enable the requested service of data. Ex.1001 ¶¶549–50.

E. Ground 5: Claims 17–18 are obvious Over Ellis in View of Spivey, Kellicker, Phillips and Pantos

1. Motivation to Combine

As discussed above, Kellicker (among others) teaches an HLS CDN caching server architecture (*e.g.*, 1280/1282) communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server (*e.g.*, 1250). Kellicker teaches using a “master rendition list” for purposes of fulfilling a particular client's streaming request. Ex.1049 ¶4(“streaming server may include the server ID in a master rendition list (*e.g.*, a manifest, a playlist, etc.)”); Ex.1001 ¶134.

Pantos is version 7 of the HLS streaming protocol, dated April 4, 2016, and originally developed by Apple in 2009. Pantos describes in detail the protocol for transferring streams of multimedia data, and specifies the format of the files and

actions to be taken by the server and clients in the transfer of the streams. Ex.1052

Abstract; Ex.1001 ¶¶135–36. Pantos teaches that the protocols described therein:

[P]rovides a reliable, cost-effective means of delivering continuous and long-form video of the Internet. It allows the receiver to adapt the bit rate of the media to the current network conditions in order to maintain uninterrupted playback at the best possible quality... It can efficiently offer multiple renditions of the same content

Ex.1052 §1.

Ellis, Spivey, Kellicker, Phillips and Pantos all involve media content delivery including real-time streaming of live or real-time media. Each describe methods and/or systems for streaming video and/or event data to one or more users. The references solve the same or similar problems, namely, providing video and/or event data to multiple users, and provide the ability to support the processing and transmission of the broadcast media utilizing several different well-known transmission protocols. Ex.1001 ¶137. A POSA would have successfully combined Ellis, Spivey Kellicker, Phillips and Pantos because such modifications involve only the substitution or addition of known elements in a predictable manner to achieve a known or desired result. Ex.1001 ¶137.

2. Dependent Claim 17

a. Elements 17.0–17.1

These elements are the same as elements 10.0–10.4. *See* §§VIII.B.2.a.–c.; Ex.1001 ¶552.

b. Element 17.2

This element is the same as elements 11.1–11.4. *See* §§VIII.C.2.a.–c.; Ex.1001 ¶553.

c. Element 17.3

This element is the same as element 15.0. *See* §VIII.C.3.a.; Ex.1001 ¶554.

d. Element 17.4

This element is the same as element 16.0. *See* §VIII.C.4.a.; Ex.1001 ¶555.

e. Element 17.5

These aspects of claim 17 refer to nothing more than the contents of a master playlist configured in accordance with known conventional HLS streaming protocol. Indeed, to support the HLS format, the master playlists, video playlists, and corresponding chunklists ***must*** conform to the HLS Specification. Ex.1052, §4(“Playlists that violate these rules are invalid; clients MUST fail to parse them.”); Ex.1001 ¶¶556–57.

For example, Kellicker describes using “a master rendition list” or “manifest” (*i.e.*, an HLS “file suite”) in order to fulfill a streaming request from a client to a server (1250) in accordance with HLS streaming protocol. Ex.1049 ¶¶4, 15, 27(“a

master rendition list, then receiving a rendition chunk list corresponding to a rendition identified by the master rendition list”); Ex.1001 ¶558.

As disclosed by Kellicker, this master rendition list may specify a first plurality of different versions (*i.e.*, renditions) of the first broadcaster's live stream of digital content. Ex.1049 ¶27. Each rendition includes a rendition chunk list and chunks (segments or fragments). *Id.* ¶¶4, 27. Kellicker teaches that this protocol applies to live streams. *Id.*

i. Playlist specifying different resolution versions

Kellicker further teaches that a plurality of renditions may be of a plurality of different versions of the same data, *i.e.*, each version having a different transmission (bit) rate. *Id.* ¶28(“When the streaming server 120 supports adaptive bitrate (ABR) streaming, the media chunks may correspond to portions of one or more ABR renditions of the VOD item 125 that are generated by the transcoder/transmuxer 126 or received from another source.”).

Similarly, Pantos recommends including resolution variants in its playlists intended to stream video. Ex.1052 §4.3.4.2. It would have been obvious to a POSA to have included resolution variants in the HLS file suite to provide adaptability and flexibility to match the appropriate resolution version of the requested data with the capabilities of the client's player device and software. Ex.1001 ¶¶559–60.

ii. Chunklist pointers to respective chunklists

As seen from the exemplary master playlist taught by Pantos and reproduced below, the HLS playlist includes a corresponding first plurality of chunklist pointers to first respective chunklists. Each URI (*e.g.*, <http://example.com/low.m3u8>, <http://example.com/mid.m3u8>, <http://example.com/hi.m3u8>) is a chunklist pointer to first respective chunklists (*e.g.*, a media playlist). Ex.1052 §4(“A Playlist is a Media Playlist if all URI lines in the Playlist identify Media Segments. A Playlist is a Master Playlist if all URI lines in the Playlist identify Media Playlists lines in the Playlist identify Media Playlists”); Ex.1001 ¶561.

These chunklist pointers point to first respective chunklists for the first plurality of different versions (*e.g.*, low bandwidth/, mid bandwidth/, and high bandwidth/).

8.4. Master Playlist

#EXTM3U

#EXT-X-STREAM-

INF:BANDWIDTH=1280000,AVERAGE-

BANDWIDTH=1000000

<http://example.com/low.m3u8>

#EXT-X-STREAM-

INF:BANDWIDTH=2560000,AVERAGE-

BANDWIDTH=2000000

<http://example.com/mid.m3u8>

#EXT-X-STREAM-

INF:BANDWIDTH=7680000,AVERAGE-

BANDWIDTH=6000000

<http://example.com/hi.m3u8>

#EXT-X-STREAM-

INF:BANDWIDTH=65000,CODECS="mp4a.40.5"

<http://example.com/audio-only.m3u8>

Ex.1052, §8.4; Ex.1001 ¶562.

iii. File segment pointers to respective file segments

As noted above, each of the chunklist pointers points to a media playlist (*e.g.*, a chunklist). The HLS Specification provides an exemplary media playlist:

#EXTM3U

#EXT-X-VERSION:3

#EXT-X-TARGETDURATION:8

#EXT-X-MEDIA-SEQUENCE:2680

#EXTINF:7.975,

<https://priv.example.com/fileSequence2680.ts>

#EXTINF:7.941,

<https://priv.example.com/fileSequence2681.ts>

#EXTINF:7.975,

<https://priv.example.com/fileSequence2682.ts>

Id. §8.2; Ex.1001 ¶563.

The exemplary media playlist includes a first plurality of file segment pointers (<https://priv.example.com/fileSequence2680.ts>, <https://priv.example.com/fileSequence2681.ts>, <https://priv.example.com/fileSequence2682.ts>) to first respective file segments ([fileSequence2680.ts](#), [fileSequence2681.ts](#), [fileSequence2682.ts](#)) for each different version of the first plurality of different versions. *Id.* §2(“A Media Playlist contains a list of Media Segments, which when played sequentially will play the multimedia presentation.”), §3(“A Media Playlist contains a series of Media Segments which make up the overall presentation. A Media Segment is specified by a URI and optionally a byte range.”); Ex.1001 ¶¶564–65.

As discussed above with respect to claims 10 and 11 (§§VIII.B.2.a.–c.; VIII.C.2.a.–c), Kellicker in view of Phillips discloses an RTMP media server and a WebRTC media server transforming live stream content into HLS format and transmitting it to a national sever at the peak of the distribution hierarchy of an HLS

CDN. The master playlist (master rendition list or HLS file suite), media playlists (chunk lists and pointers), and segment files (and pointers) describe the components that are required by the HLS protocol itself in order for clients to play the media. For this reason alone, their inclusion in the master rendition list (HLS file suite) would have been obvious to a POSA. Ex.1001 ¶566. Furthermore, configuring the master rendition list (HLS file suite) in this manner serves to guide the media player on how to retrieve and play video content by providing details about each chunk or segment and allowing for adaptive streaming based on network conditions, as well as the resolution capabilities of the user's media player. Ex.1001 ¶¶567–68.

f. Element 17.6

This element is disclosed as part of the CDN distribution process and architecture taught by Phillips. §VIII.B.2.b; Ex.1050 ¶31, FIG. 1, Abstract(“a manifest available to the content delivery node with respect to a particular content as well as segment files stored in a database cache of the content delivery node are analyzed to determine if any segment files referenced by the manifest are absent from the database cache.... the absent segment files, they may be pulled from another content delivery node of the CDN (*e.g.*, a parent node or an edge server node).”). Ex.1001 ¶¶569–70. As would have been understood by a POSA, the national server of Phillips can act as the primary server where the served content is stored (first copy), which then distributes cached copies of that content from

geographically distributed edge servers closer to users for faster access (second copy). Ex.1001 ¶571; §VIII.C.3.a.

The inclusion of playlists specifying different resolution versions, chunklists and chunklists pointers, as well as segments and segment pointers in the master rendition list (*i.e.*, HLS file suite) is rendered obvious by Kellicker as modified by Pantos. §VIII.E.2.e.

g. Element 17.7

This element is satisfied for the reasons explained in §§VIII.C.2.a.–c., VIII.C.3.a., VIII.E.2.b–f.; Ex.1001 ¶572; Ex.1050 ¶¶8–9, 31, 59.

h. Element 17.8

Again, this element is satisfied for the reasons explained in §§VIII.C.2.a.–c.; VIII.C.3.a., VIII.E.2.b–f.; Ex.1001 ¶573; Ex.1050 ¶¶8–9, 31, 59.

3. Claim 18

a. Elements 18.0–18.1

Pantos teaches that segments contained in HLS playlists have a specified target duration. Ex.1052 §4.3.2.1(“The EXTINF tag specifies the duration of a Media Segment... This tag is REQUIRED for each Media Segment.”). Thus, as was readily understood by a POSA, the server must be “configurable” to set and include a video segment duration in the HLS file suite in order to comply with the HLS protocol. Thus, this element would have been obvious to a POSA for this reason alone. Ex.1001 ¶¶574–75.

b. Element 18.2

As explained in the HLS Specification, the number of file segment pointers in a chunklist contained a media playlist may be limited. Ex.1052, §6.2.2(“the server MAY limit the availability of Media Segments by removing Media Segments from the Playlist file. If Media Segments are to be removed, the Playlist file MUST contain an EXT—MEDIA—SEQUENCE tag.”). Thus, as was readily understood by a POSA, the server must be “configurable” to set and include a buffer length including a number of file segments, and their associated pointers, and each chunk list in the HLS file suite in order to comply with the HLS protocol. Thus, this element would have been obvious to a POSA. Ex.1001 ¶¶576–77.

c. Element 18.3

Pantos discloses, by way of example, a video segment duration of “less than 10 seconds” or “3.003 seconds.” Ex.1052, §2. Two seconds is both less than 10 seconds and “approximately” 3.003 seconds. Therefore, a POSA would have been motivated to select a video segment duration of approximately two seconds, consistent with the HLS protocol, and as appropriate for live streaming content. Ex.1001 ¶¶578–79.

d. Element 18.4

As explained in the HLS specification, “the server MUST NOT remove a Media Segment from a Playlist file without an EXT—X—ENDLIST tag if that would produce a Playlist whose duration is less than three times the target duration. Doing

so can trigger playback stalls.” Ex.1052 §6.2.2. Accordingly, the absolute minimum number of file segment pointers would be three (*i.e.*, three times the target duration). Therefore, the claimed four, five or six pointers (and associated file segments) represent values consistent with the HLS Specification. Therefore, a POSA would have been motivated to select a plurality of file segments (and associated pointers) in each chunk list for the HLS buffer length that is 4, 5 or 6, consistent with the HLS protocol, and in order to avoid playback stalls. Ex.1001 ¶¶580–81.

IX. SECONDARY CONSIDERATIONS OF NON–OBVIOUSNESS

Petitioner is aware PO has made conclusory assertions of secondary considerations in its Preliminary Responses filed in the Genius IPRs. However, PO clearly fails to establish a nexus or a presumption of a nexus. Thus, PO's conclusory assertions in this regard are unavailing. Ex.1001 ¶¶690. Moreover, assuming PO were to present similar arguments with its Preliminary Response in this case, Petitioner would not have the opportunity to respond. Thus, a decision on the adequacy of such evidence at the institution phase without a fully developed record, is impracticable. Indeed, the number of “[i]ndependently made, simultaneous inventions, made within a comparatively short space of time, are persuasive” “indicia of obviousness.” *Geo. M. Martin v. Alliance Machine*, 618 F.3d 1294, 1304–1305 (Fed. Cir. 2010) (quotations omitted).

X. THE BOARD SHOULD NOT EXERCISE DISCRETIONARY DENIAL

A. The '687 Has Not Been Subject to a Prior Petition by the Current Petitioner

Furthermore, there is no “significant relationship” between the prior Petitioner (Genius) and the present Petitioner (Sportradar). *Videndum Production Solutions, Inc. v. Rotolight Limited*, IPR2023–01218, Paper 12 (April 19, 2024). *See also, Petitioner's Reasons for Parallel Petitions and Petition Ranking* filed concurrently herewith.

B. *Fintiv*

Petitioner has stipulated to PO that if this IPR is instituted, Petitioner will not pursue invalidity on the grounds raised here or on any other ground that reasonably could have been raised in this IPR. Ex.1021. Thus, the Board should not exercise discretionary denial. Ex.1013.

XI. CONCLUSION

For the foregoing reasons, Petitioner respectfully requests that the Board institute IPR and cancel the Challenged Claims.

Petition for *Inter Partes* Review of U.S. Patent No. 10,805,687

Date: December 14, 2024

Respectfully submitted,

/Scott W. Cummings/

Scott W. Cummings

Reg. No. 41,567

BRYAN CAVE LEIGHTON PAISNER LLP

scott.cummings@bclplaw.com

Alexander Walden

Provisionally recognized PTAB attorney
(affidavit to follow)

BRYAN CAVE LEIGHTON PAISNER LLP

alexander.walden@bclplaw.com

Frank Fabiani

Reg. No. 75,934

BRYAN CAVE LEIGHTON PAISNER LLP

Attorneys For Petitioner Sportradar AG

CERTIFICATE OF COMPLIANCE

This Petition complies with the type–volume limitations as mandated in 37 C.F.R. §42.24. According to the word processing system used to prepare this document, the brief contains 13,925 (14,000 limit) words.

December 14, 2024

/Scott W. Cummings/

Scott W. Cummings

Reg. No. 41,567

BRYAN CAVE LEIGHTON PAISNER LLP

scott.cummings@bclplaw.com

CERTIFICATE OF SERVICE

In compliance with 37 C.F.R. §§42.105, 42.6(e), the undersigned hereby certifies that a copy of the foregoing Petition, supporting exhibits, and Power of Attorney were served on the 14th day of December 2024, via Federal Express® directed to PO at the correspondence address of record:

149890 – Smith Baluch LLP
376 Boylston St.
Suite 401
Boston, MA 02116

A courtesy copy was also served by electronic mail on PO's counsel of record in the Texas litigation:

Britton F. Davis
KING & SPALDING LLP
1401 Lawrence Street
Suite 1900
Denver, CO 80202
(720) 535-2300
bfdavis@kslaw.com

Christopher C. Campbell
KING & SPALDING LLP
1700 Pennsylvania Avenue, NW
Suite 200
Washington, DC 20006
(202) 626-5578
ccampbell@kslaw.com

Ryan A. Schmid
KING & SPALDING LLP
1700 Pennsylvania Avenue, NW
Suite 200
Washington, DC 20006
(202) 626-5578
rschmid@kslaw.com

Matthew D. Wood
KING & SPALDING LLP
500 West 2nd Street
Suite 1800
Austin, TX 78701
(512) 457-2000
mwood@kslaw.com

/Teresa C. Rodriguez/

TABLE OF EXHIBITS⁷

Exhibit No.	Description
Exhibit 1001	Declaration of Dr. Omid Kia, Ph.D.
Exhibit 1002	<i>Curriculum Vitae</i> of Dr. Omid Kia, Ph.D.
Exhibit 1003	U.S. Patent No. 10,805,687
Exhibit 1004	Claim Listing of U.S. Patent No. 10,805,687
Exhibit 1005	Certified File History of U.S. Patent No.10,805,687
Exhibit 1006	U.S. Patent No. 10,425,697
Exhibit 1007	Claim Listing of U.S. Patent No. 10,425,697
Exhibit 1008	Certified File History of U.S. Patent No. 10,425,697
Exhibit 1009	U.S. Patent No. 11,039,218
Exhibit 1010	RESERVED
Exhibit 1011	Certified File History of U.S. Patent No. 11,039,218
Exhibit 1012	RESERVED
Exhibit 1013	Director Vidal Memorandum re: <i>Sotera</i> Stipulations
Exhibit 1014	SportsCastr's Complaint for Patent Infringement against Sportradar Group AG
Exhibit 1015	U.S. Patent Pub. No. 2015/0163379 ("Herzog")
Exhibit 1016	U.S. Patent Pub. No. 2014/0229992 ("Ellis")

⁷ Brackets ([]) are used to refer to the sequential page numbers added to exhibits pursuant to 37 C.F.R. §42.63(d)(2)(i).

Exhibit 1017	U.S. Patent Pub. No. 2016/0036910 (“Spivey”)
Exhibit 1018	U.S. Patent Pub. No. 2015/0163379 (“Mitic”)
Exhibits 1019–1020	RESERVED
Exhibit 1021	Sportradar’s proposed <i>Sotera</i> Stipulation to SportsCastr
Exhibit 1022	James F. Kurose & Keith W. Ross, <i>Application Layer</i> , in Computer Networking a Top–Down Approach (Michael Hirsch <i>et al.</i> eds., 4th ed. 2012) (“Kurose”)
Exhibit 1023	John Dilley <i>et al.</i> , <i>Globally Distributed Content Delivery</i> , IEEE Internet Computing (Sept.–Oct. 2002).
Exhibit 1024	Mário Marques da Silva, <i>Multimedia Communications and Networking</i> (2012) (“Silva”)
Exhibit 1025	<i>A Brief History of the Internet</i> , ONLINE LIB. LEARNING CENTER. (Feb. 9, 2024), https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml (“Online Library”)
Exhibit 1026	Hisayuuki Ohmata <i>et al.</i> , <i>Hybridcast: A new media experience by integration of broadcasting and broadband</i> , 2013 ITU Kaleidoscope: Building Sustainable Communities, April 2013 (“Ohmata”)
Exhibit 1027	Henning Schulzrinne <i>et al.</i> , <i>RTP: A Transport Protocol for Real–Time–Applications</i> , DataTracker (January 1996), https://datatracker.ietf.org/doc/html/rfc1889 (“Schulzrinne”)
Exhibit 1028	Brijesh Kumar, <i>Making Sense of Video Streaming Protocols</i> , LinkedIn. (Aug. 11, 2016), https://www.linkedin.com/pulse/making-sense-video-streaming-protocols-dr-brijesh-kumar/ (“Kumar”)
Exhibit 1029	Ian Fette & Alexey Melnikov, <i>The WebSocket Protocol</i> , RFC Editor. (Dec. 2016), https://www.rfc-editor.org/rfc/rfc6455.html .
Exhibit 1030	Marcin Warczygłowa, <i>Real–time Web Application with Websockets and Vert.X</i> (Nov. 10, 2015), https://blog.allegro.tech/2015/11/real-time-web-application-with-websockets-and-vert-x.html .

Exhibit 1031	Randall Stewart & Chris Metz, <i>SCTP: new transport protocol for TCP/IP</i> , 5 IEEE 64, (2001) (“Stewart”)
Exhibit 1032	Lydon Ong, John Yoakum &, <i>An Introduction to the Stream Control Transmission Protocol (SCTP)</i> , Datatracker. (May. 2002), https://datatracker.ietf.org/doc/rfc3286/ (“Ong”)
Exhibit 1033	Jan Newmarch, <i>Introduction to Stream Control Transmission Protocol</i> , LINUX J. (Sep. 1, 2007), https://www.linuxjournal.com/article/9748 (“Newmarch”)
Exhibit 1034	Kyung-Hoe Kim et al, <i>A transmission control SCTP for real-time multimedia streaming</i> , 54 Computer Networks 1418, (2010) (“Kim”)
Exhibit 1035	High Scalability, <i>How Facebook Live Streams to 800,000 Simultaneous Viewers</i> , ByteByteGo (Jun. 27, 2016), https://blog.bytebytego.com/?utm_source=hs&utm_medium=ann&utm_campaign=top (“ByteGo”)
Exhibit 1036	Hardeep Singh Parmar & Michael C. Thornburgh, <i>Adobe’s Real Time Messaging Protocol</i> , Veriskope. (Dec. 21, 2002) (“Parmer”)
Exhibit 1037	Ilya Grigorik, <i>WebRTC</i> , in <i>High Performance Browser Networking</i> , (2013) (“Grigorik”)
Exhibit 1038	Sam Dutton, <i>Get started with WebRTC</i> , web.dev. (Jul. 23, 2012), https://web.dev/articles/webrtc-basics (“Dutton”)
Exhibit 1039	Andy Salo, <i>MPEG DASH: A Technical Deep Dive and Look at What’s Next</i> , RGB Networks, (2012) (“Salo”)
Exhibit 1040	Jan Ozer, <i>Upstream, Justin.tv, Livestream, and Bambuser: Streaming Unplugged</i> , Streaming Media Magazine., Jun.–Jul. 2012 (“Ozer”)
Exhibit 1041	Christopher Mueller, <i>MEPG–DASH vs. Apple HLS vs. Microsoft Smooth Streaming vs. Adobe HDS</i> , BITMOVIN. (Mar. 29, 2015), https://bitmovin.com/mpeg-dash-vs-apple-hls-vs-microsoft-smooth-streaming-vs-adobe-hds/ (“Mueller”)
Exhibit 1042	Stefan Saroiu et al., <i>An analysis of Internet Content Delivery Systems</i> , ACM SIGOPS Operating System Review, Vol. 36 (Dec. 31, 2002), https://doi.org/10.1145/844128.84415 (“Saroiu”)

Petition for *Inter Partes* Review of U.S. Patent No. 10,805,687

Exhibit 1043	Charles Marion & Julien Jomier, <i>Real-time Collaborative Scientific WebGL Visualization with Websocket</i> , Ass’n Computing Machinery, August 4–5, 2012 (“Marion”)
Exhibit 1044	Mark Sweney, <i>First ads appear on YouTube Clips</i> , The Guardian, Aug. 22, 2007 (“Sweney”)
Exhibit 1045	Richard Sandomir, <i>The Innovation That Grew and Grew</i> , N.Y. Times, June 12, 2014 (“Sandomir”)
Exhibit 1046	Victoria Pimentel & Bradford G. Nickerson, <i>Communicating and Displaying Real-Time Data with WebSocket</i> , 16 IEEE (2012) (“Pimentel”)
Exhibit 1047	Lucian Popa <i>et al.</i> , <i>HTTP as the narrow waist of the future internet</i> , 2010 ACM SIGCOMM Workshop, October 2013 (“Popa”)
Exhibit 1048	Gilstrap Order of September 30, 2024 authorizing addition of Challenged Claims to the Complaint
Exhibit 1049	U.S. Patent Publication No. 2017/0250882 (“Kellicker”)
Exhibit 1050	U.S. Patent Publication No. 2015/0127845 (“Phillips”)
Exhibit 1051	U.S. Patent No. 9,246,965 (“Stoica”)
Exhibit 1052	R. Pantos <i>et al.</i> , “HTTP Live Streaming,” (“Pantos”) published April 4, 2016
Exhibit 1053	Wayback Machine Pantos archive dated April 4, 2016
Exhibit 1054	US Patent No. 9,106,607, (“Lepeska”)
Exhibit 1055	EP2920708 (“Basile”)
Exhibit 1056	U.S. Patent No. 9,973,785 (“Yang”)