UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SPORTRADAR GROUP AG,
Petitioner

v.

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

U.S. Patent Nos. 10,425,697 and 10,805,687

DECLARATION OF DR. OMID KIA IN SUPPORT OF PETITIONS FOR *INTER PARTES* REVIEW OF U.S. PATENT NOS. 10,425,697 AND 10,805,687

TABLE OF CONTENTS

Page

I. Introduction....1

II. Qualifications and Background....3

A. Education Background and Career History....3

B. Publications and Patents....8

C. Prior Testimony....10

III. Understanding of Patent Law....10

IV. Description of the Technology....13

A. Internet Communications in General....13

B. Network Communication Models....18

C. Protocols for Real-time or Low-latency Data Transmission....20

1. Real-Time Transport Protocol (RTP)....20

2. HTTP Polling....20

3. WebSocket....21

4. Stream Control Transmission Protocol (SCTP)....22

5. Real-Time Messaging Protocol (RTMP)....22

6. Web Real-Time Communication (WebRTC)....23

7. Dynamic Adaptive Streaming Over HTTP ("DASH" or "MPEG-DASH") / HTTP Live Streaming (HLS)....24

D. Internet Live Video Streaming....25

V. Overview of the Challenged Patents....26

A. Alleged Problem in the Art....26

B. Alleged Invention of the Challenged Patents....27

VI. Level of Ordinary Skill in the Art....32

VII. Claim Construction....34

A. The '687 and '697 Patents....34

VIII. Summary of the Prior Art....35

A. U.S. Patent Pub. No. 2015/0163379 ("Herzog")....35

B. U.S. Patent Pub. No. 2014/0229992 ("Ellis")....39

C. U.S. Patent Pub. No. 2016/0036910 ("Spivey")....42

IX. A POSA's Motivation To Combine Prior Art References with a Reasonable Expectation of Success....48

A. Ellis in View of Spivey....48

1. *Ellis*....48

2. Socket Server (Spivey)....49

B. Ellis in View of Spivey and Kellicker....55

C. Ellis in View of Spivey, Kellicker, and Phillips....58

D. Ellis in View of Spivey, Kellicker, Phillips, and Stoica....62

E. Ellis in View of Spivey, Kellicker, Phillips, and Pantos....66

F. Herzog in View of Spivey and Ellis....68

1. Socket server (*Spivey*)....70

2. Providing Users with Chat, Sports Score, and Sports Commentary Information Using Separate Communication Channels (Ellis)....72

G. Herzog in View of Spivey, Ellis, Kellicker, and Phillips....77

X. Grounds....83

A. The '697 86

1. '697 Ground 1: Claim 1 is Obvious Over Ellis in View of Spivey 86

2. '697 Ground 2: Claim 10 is Obvious Over Ellis in View of Spivey and Kellicker 113

3. '697 Ground 3: Claims 11 and 15-16 are Obvious Over Ellis in View of Spivey, Kellicker, and Phillips 119

4. '697 Ground 4: Claims 12-14 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips, and Stoica 134

5. '697 Ground 5: Claims 17-18 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips, and Pantos 153

6. '697 Ground 6: Claims 23 and 24 Are Obvious Over Ellis in View of Spivey 169

7. '697 Ground 7: Claim 26 is Obvious Over Ellis in View of Spivey, Kellicker, and Phillips 193

8. '697 Ground 8: Claims 23-24 Are Obvious Over Herzog in View of Spivey and Ellis 205

9. '697 Ground 9: Claim 26 is Obvious Over Herzog in View of Spivey, Ellis, Kellicker, and Phillips 227

B. The '687 238

1. '687 Ground 1: Claim 1 is Obvious Over Ellis in View of Spivey 238

2. '687 Ground 2: Claim 10 is Obvious Over Ellis in View of Spivey and Kellicker 263

3. '687 Ground 3: Claims 11 and 15-16 are Obvious Over Ellis in View of Spivey, Kellicker, and Phillips 269

4. '687 Ground 4: Claims 12-14 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips, and Stoica 284

5. '687 Ground 5: Claims 17-18 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips, and Pantos.....303

6. '687 Ground 6: Claims 23 and 24 Are Obvious Over Ellis in View of Spivey.....318

7. '687 Ground 7: Claim 26 is Obvious Over Ellis in View of Spivey, Kellicker, and Phillips.....340

8. '687 Ground 8: Claims 23-24 Are Obvious Over Herzog in View of Spivey and Ellis.....352

9. '687 Ground 9: Claim 26 is Obvious Over Herzog in View of Spivey, Ellis, Kellicker, and Phillips.....369

XI. Secondary Considerations of Non-Obviousness.....381

XII. Conclusion.....382

TABLE OF EXHIBITS[1]

Exhibit No.	Description
Exhibit 1001	Declaration of Dr. Omid Kia, Ph.D. (*i.e.*, this declaration)
Exhibit 1002	*Curriculum Vitae* 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: *Sotera* 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")

[1] Unless otherwise specified, citations are to the original page, column, and line numbers in exhibits. 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")
Exhibit 1019-20	RESERVED
Exhibit 1021	Sportradar's proposed *Sotera* Stipulation to SportsCastr
Exhibit 1022	James F. Kurose & Keith W. Ross, *Application Layer*, in Computer Networking A Top-Down Approach (Michael Hirsch et al. eds., 4th ed. 2012) ("Kurose")
Exhibit 1023	John Dilley et al., *Globally Distributed Content Delivery*, IEEE Internet Computing (Sept.–Oct. 2002).
Exhibit 1024	Mário Marques da Silva, Multimedia Communications and Networking (2012) ("Silva")
Exhibit 1025	*A Brief History of the Internet*, ONLINE LIB. LEARNING CENTER. (Feb. 9, 2024), https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml ("Online Library")
Exhibit 1026	Hisayuuki Ohmata et al., *Hybridcast: A new media experience by integration of broadcasting and broadband*, 2013 ITU Kaleidoscope: Building Sustainable Communities, April 2013 ("Ohmata")
Exhibit 1027	Henning Schulzrinne et al., *RTP: A Transport Protocol for Real-Time-Applications*, DataTracker (January 1996), https://datatracker.ietf.org/doc/html/rfc1889 ("Schulzrinne")
Exhibit 1028	Brijesh Kumar, *Making Sense of Video Streaming Protocols*, LinkedIn. (Aug. 11, 2016), https://www.linkedin.com/pulse/making-sense-video-streaming-protocols-dr-brijesh-kumar/ ("Kumar")
Exhibit 1029	Ian Fette & Alexey Melnikov, *The WebSocket Protocol*, RFC Editor. (Dec. 2016), https://www.rfc-editor.org/rfc/rfc6455.html.

Exhibit 1030	Marcin Warczygłowa, *Real-time Web Application with Websockets and Vert.X* (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, *SCTP: new transport protocol for TCP/IP*, 5 IEEE 64, (2001) ("Stewart")
Exhibit 1032	Lydon Ong, John Yoakum &, *An Introduction to the Stream Control Transmission Protocol (SCTP)*, Datatracker. (May. 2002), https://datatracker.ietf.org/doc/rfc3286/ ("Ong")
Exhibit 1033	Jan Newmarch, *Introduction to Stream Control Transmission Protocol*, LINUX J. (Sep. 1, 2007), https://www.linuxjournal.com/article/9748 ("Newmarch")
Exhibit 1034	Kyung-Hoe Kim et al, *A transmission control SCTP for real-time multimedia streaming*, 54 Computer Networks 1418, (2010) ("Kim")
Exhibit 1035	High Scalability, *How Facebook Live Streams to 800,000 Simultaneous Viewers*, 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, *Adobe's Real Time Messaging Protocol*, Veriskope. (Dec. 21, 2002) ("Parmer")
Exhibit 1037	Ilya Grigorik, *WebRTC*, in High Performance Browser Networking, (2013) ("Grigorik")
Exhibit 1038	Sam Dutton, *Get started with WebRTC*, web.dev. (Jul. 23, 2012), https://web.dev/articles/webrtc-basics ("Dutton")
Exhibit 1039	Andy Salo, *MPEG DASH: A Technical Deep Dive and Look at What's Next*, RGB Networks, (2012) ("Salo")
Exhibit 1040	Jan Ozer, *Upstream, Justin.tv, Livestream, and Bambuser: Streaming Unplugged*, Streaming Media Magazine., Jun.–Jul. 2012 ("Ozer")

Exhibit 1041	Christopher Mueller, *MEPG-DASH vs. Apple HLS vs. Microsoft Smooth Streaming vs. Adobe HDS*, 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., *An analysis of Internet Content Delivery Systems*, ACM SIGOPS Operating System Review, Vol. 36 (Dec. 31, 2002), https://doi.org/10.1145/844128.84415 ("Saroiu")
Exhibit 1043	Charles Marion & Julien Jomier, *Real-time Collaborative Scientific WebGL Visualization with Websocket*, Ass'n Computing Machinery, August 4-5, 2012 ("Marion")
Exhibit 1044	Mark Sweney, *First ads appear on YouTube Clips*, The Guardian, Aug. 22, 2007 ("Sweney")
Exhibit 1045	Richard Sandomir, *The Innovation That Grew and Grew*, N.Y. Times, June 12, 2014 ("Sandomir")
Exhibit 1046	Victoria Pimentel & Bradford G. Nickerson, *Communicating and Displaying Real-Time Data with WebSocket*, 16 IEEE (2012) ("Pimentel")
Exhibit 1047	Lucian Popa et al., *HTTP as the narrow waist of the future internet*, 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 et al., "HTTP Live Streaming," ("Pantos") published April 4, 2016
Exhibit 1053	Wayback Machine Pantos archive dated April 4, 2016

Exhibit 1054	U.S. Patent No. 9,106,607) (“Lepeska”)
Exhibit 1055	EP2920708 (“Basile”)
Exhibit 1056	U.S. Patent No. 9,973,785 (“Yang”)

Sportradar AG Ex. 1001 - Declaration of Dr. Omid Kia, Ph.D.

I, Dr. Omid Kia, declare as follows:

I. INTRODUCTION

1. My name is Omid Kia, Ph.D. I have been retained on behalf of Sportradar AG ("Sportradar") to provide my opinions regarding whether claims 1, 10-18, 23-24 and 26 of U.S. Patent No. 10,805,687 ("the '687 patent") and claims 1, 10-18, 23-24 and 26 of U.S. Patent No. 10,425,697 ("the '697 patent") (together "the Challenged Claims" and "the Challenged Patents," respectively) would have been invalid to one of ordinary skill in the art as of August 5, 2016 in the above-captioned Petitions for *Inter Partes* Review ("IPRs") of the '687 and'697 patents. In a related set of proceedings I have also provided opinions in support of six Petitions for *Inter Partes* Review ("IPRs") regarding whether claims 1-9, 19-25, and 27-30 of U.S. Patent No. 10,805,687 ("the '687 patent"); claims 1-9, 19-25, 27-30 of U.S. Patent No. 10,425,697 ("the '697 patent"); and claims 1-30 of U.S. Patent No. 11,039,218 ("the '218 patent") would have been invalid to one of ordinary skill in the art as of August 5, 2016.

2. For time spent in connection with this matter, I will be compensated at my standard billing rate of $675 per hour. I am being separately reimbursed for any out-of-pocket expenses. My compensation does not depend on the content of my opinions or the outcome of this case.

3. I understand that the previous six IPRs filed by Sportradar raise the same prior art and the same grounds as six IPRs previously filed by Genius Sports Ltd. ("Genius") against the '687, '697, and '218 patents, namely: (i) for the '697 patent, IPR2024-01308 (claims 1-9) and IPR2024-01309 (claims 19-25, 27-30), (ii) for the '687 patent, IPR2024-01305 (claims 1-9) and IPR2024-0107 (claims 19-25, 27-30), and (iii) for the '218 patent, IPR2024-01310 (claims 1-15) and IPR2024-01311 (claims 16-30) (collectively, the "Genius IPRs"). I also understand that, in support of the Genius IPRs, Genius submitted the declaration of Dr. Stephen B. Wicker ("Wicker Declaration," attached hereto as Exhibit 1048). In forming my substantive opinions for this declaration I have also considered the substantive opinions, testimony, and evidence cited in the Wicker Declaration, labeled as Exhibit 1001 in the Genius IPRs.

4. In forming the opinions expressed in this Declaration, I have also relied on my education, training, research, knowledge, and my academic and personal and professional experience. I also considered the viewpoint of a person having ordinary skill in the art as of the time of alleged invention of the '687 and '697 patents. My opinions are based, at least in part, on the following:

- U.S. Patent No. 10,805,687 ("the '687 Patent") (Ex. 1003)
- U.S. Patent No. 10,425,697 ("the '697 patent") (Ex. 1006)
- U.S. Patent No. 11,039,218 ("the '218 Patent") (Ex. 1009)

- U.S. Patent No. 10,805,687 Certified File History (Ex. 1005)
- U.S. Patent No. 10,425,697 Certified File History (Ex. 1008)
- U.S. Patent No. 11,039,218 Certified File History (Ex. 1011)
- U.S. Patent Pub. No. 2015/0163379 ("Herzog") (Ex. 1015)
- U.S. Patent Pub. No. 2014/0229992 ("Ellis") (Ex. 1016)
- U.S. Patent Pub. No. 2016/0036910 ("Spivey") (Ex. 1017)
- U.S. Patent Pub. No. 2017/0250882 ("Kellicker") (Ex. 1049)
- U.S. Patent Pub. No. 2015/0127845 ("Phillips") (Ex. 1050)
- U.S. Patent No. 9,246,965 ("Stoica") (Ex. 1051)
- R. Pantos et al., "HTTP Live Streaming," ("Pantos") (Ex. 1052)
- U.S. Patent No. 9,106,607) ("Lepeska") (Ex. 1054)
- EP2920708 ("Basile") (Ex. 1055)
- U.S. Patent No. 9,973,785 ("Yang") (Ex. 1056)

II. QUALIFICATIONS AND BACKGROUND

A. Education Background and Career History

5. My qualifications for forming the opinions set forth in this report are summarized in this section and in my CV, attached hereto as Exhibit 1002. My CV provides an accurate identification of my background and experience, as well as a list of my publications.

6. I earned my Bachelor's of Science in electrical engineering from Catholic University America in 1987. In 1989, I completed my Master's of Science in electrical engineering from the University of Illinois at Chicago. I then attended

the University of Maryland at College Park, where I earned my Ph.D. in electrical engineering in 1997.

7. A chief focus of my career has been designing and developing computed imaging, video, and content processing systems for medical and other uses, considering all aspects of the user, patient and system components, including the requisite hardware, software, networking, and user-facing system elements. My experience in this space spans industry and academia. For example, after I completed my Ph.D., I took the position of Chief Technical Officer at IMACOM in 1999. In this role, I managed a team of engineers in developing medical imaging software – including both ultrasound and X-ray imaging applications. Among other things, my role involved the development of video pipeline processing, real-time storage and retrieval of live video, creating custom software applications and integrating custom graphical user interfaces to meet customer demands. For example, I designed algorithms where the user interface exploits the underlying information by interacting with the user interface implemented in desktop, server and mobile environments.

8. With respect to the subject matter of the ’687 and ’697 patents addressed in this Declaration, I have over 25 years of experience in design and development of complex and network-based imaging systems for various purposes including video, consumer, medical, and portable imaging systems amongst many

other modalities. My experience includes design and development of systems for video and content processing and communication, among many other concepts, including Video over IP, video surveillance, storage and processing of real-time live video, multimedia compression and communication, and image and video indexing. For example, after I completed my Ph.D., I worked as an Electronics Engineer at the National Institute of Standard and Technology, where I worked on various projects involving multimedia compression and communication and video indexing using media processing tools to develop new and innovative techniques for archiving, processing and transmitting video and other media. I also researched techniques for serving large microscope images annotated with educational pointers for educational purposes and served as the U.S. Ambassador to the MPEG standardization Group working on the standards for MPEG-4 at the time. As another example, while working as a principal consultant for Sigma Vision, I authored several proposals directed to Video Over IP and video surveillance systems, including development of Video Over IP systems for thin clients that were java enabled. Later on, as President of Sigma Vision, I also performed several large scale consulting projects in the area of Video Over IP.

9. I am currently a senior systems engineer, serving as subject matter expert at Coastal Communication Consultants, Inc. (hereinafter referred to as "CCCi"). In this capacity, I serve as an expert in all of CCCi's imaging and video

data initiatives along with high-technology research and development. I also serve as an expert in all of CCCi's activities pertaining to video processing, video surveillance, image/signal processing, and other software development problems.

10. Prior to joining CCCi, I served as the Chief Image Scientist at Northstrat Inc. hereinafter referred to as "Northstrat") and Senior Staff Scientist at ITT Exelis, Space Sciences Division (hereinafter referred to as "ITT Exelis") in a similar capacity. In my role at ITT Exelis and similarly Northstrat, I led research and development efforts with respect to all high technology development, including in the areas of video processing, compression and standards.

11. ITT Exelis and Northstrat expanded their business into product-based services and as such utilized my innovative approach to develop high technology products that carried a service-based revenue model. ITT Exelis (now part of Harris Corporation) was a leader in Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance ("C4ISR") related products and systems and information and technical services, supplying military, government, and commercial customers in the United States and globally. ITT Exelis was a pure-play aerospace, defense, and information solutions company with strong positions in enduring and emerging global markets, some 20,000 employees, and 2010 revenues of about $6 billion.

12. Also at ITT Exelis, I served as one of the leading experts in image and computer vision related activities. In particular I designed the camera processing for a large field of view, multi-sensor camera with field-steerable mirrors to be used in an Unmanned Aerial Vehicle (UAV). I also served as the video processing subject matter expert to address all aspects of video coding and decoding as it pertains to image quality, channel model, display, format, indexing, error recovery and transmission.

13. Prior to joining ITT, I served as the Chief Scientist at Imaging Sciences International, Inc. where my efforts catapulted the company into the leading Cone Beam Computed Tomography (CBCT) provider in the world. The application of CBCT was targeted for the general dental market with first adopters including Implantologists but later to other providers. I also served as the primary Intellectual Property Manager for all patent related work for Imaging Sciences and its affiliates, as well as a subject matter expert for all efforts involving digital imaging. Prior to joining Imaging Sciences, in my roles at Sigma Vision, Portis, and IMACOM, I researched and developed numerous medical and consumer technologies, involving imaging and signal processing, video processing, Video On Demand, video surveillance, and multimedia communication systems. These activities included hardware and software design throughout the entire system development cycle, including networking and user interface solutions.

14. Also, prior to that, and immediately after my graduation, I continued work in media compression and processing at the Compression Group in the Information Technology Laboratory of the National Institutes of Standards and Technology. In this role, I continued my research and expanded on similar topics across several media forms. In particular, I served as the United States Government ambassador to the MPEG standardization group. I also expanded on my thesis research topic to exploit compressed-domain processing of media for images, video and multimedia content. I worked with engineers who utilized the MPEG standardization body's source code for performance, quality and testing.

15. In these capacities, I have worked on many aspects of video and content systems, including algorithm development, software development, hardware development, networking architecture, and system design. I have also implemented video processing baselines for various products including performance criteria such as video quality, processing and communication requirements.

B. Publications and Patents

16. Following my Ph.D., I authored approximately 30 publications, including journal and conference papers, many of which focus on multimedia communication systems, video and image processing, and content networks and services.

17. Examples of my publications include:

- H. Li, D. Doermann and O. Kia. Text Extraction, Enhancement and OCR in Digital Videos. *Book chapter in Springer Verlag, 1999.*

- O. Kia, J. Sauvola and D. Doermann. Network-diffused media scaling for multimedia content services. *In Interactive Distributed Multimedia Systems and Telecommunication Services,* Eds. M. Diaz, P. Owezarski and P. Senac. In proceedings of the *International Workshop on Interactive Distributed Multimedia Systems and Telecommunications Services, pp. 149-162, 1999.*

- H. Li, O. Kia and D. Doermann. Text Extraction and Recognition in Digital Video. In the proceedings of the *International Workshop on Document Analysis Systems,* pp 119-128, 1998.

- H. Li, D. Doermann and O. Kia. Automatic Text Extraction and Tracking in Digital Video. *LAMP Technical Report,* LAMP-TR-028, CAR-TR-900, CS-TR-3962, Dec. 1998

- H. Li, O. Kia and D. Doermann. Text Enhancement in Digital Video. In the proceedings of the SPIE — Document Recognition and Retrieval, volume 3651, pp 2-9, 1999.

- O. Kia and J. Sauvolla. Active documents for mobile services. In the proceedings of the IEEE workshop on Multimedia Signal Processing, pp. 227-232, 1998.

- O. Kia and A. Schaff. Data Representation and Handling for Large Image Browsing. In the proceedings of the *SPIE – Multimedia Storage and Archiving Systems,* pp. 37-46, 1998

- J. Sauvola and O. Kia. Hyperdocument management for compression, transmission and processing. In Proceedings of the *IEEE Workshop on Multimedia Signal Processing,* pp. 537-542, 1997

18. A complete list of my publications is contained in my CV, a copy of which is provided as Ex.1002 to the Petition.

19. I am also a named inventor on approximately 30 patents, a complete list of which is contained in my CV

C. Prior Testimony

20. A listing of the cases in which I have testified as an expert at trial or by deposition within the preceding five-plus years are also contained in my CV.

III. UNDERSTANDING OF PATENT LAW

21. I understand that prior art to the ’687 and’697 patents includes patents and printed publications in the relevant art that predate the priority date of the ’687 and ’697 patents.

22. I understand that claims in an IPR are construed under the case *Phillips v. AWH Corp.*, 415 F.3d 1303, decided by the Federal Circuit in 2005. I understand that under the rule in *Phillips*, words of claims are given their plain and ordinary meaning as understood by a person of ordinary skill in the art in view of the specification and prosecution history, unless those sources show an intent to depart from such meaning. No terms need to be construed for the purposes of this declaration. I reserve the right to respond to any constructions offered by the Patent Owner or adopted by the Board.

23. I have also been advised that statements of intended use or intended functionality may not serve to further limit a claimed invention, especially with respect to a claimed apparatus or system.

24. I understand that in IPR a claim is unpatentable if it is anticipated or obvious. Anticipation of a claim requires that every element of a claim be disclosed expressly or inherently in a single prior art reference, arranged in the prior art reference as arranged in the claim. An element is inherent if it is necessarily present in the reference.

25. It is my understanding that a patent claim is unpatentable for obviousness if the claimed invention would have been obvious at the time the invention was made to a person having ordinary skill in the art (POSA).

26. I understand that the following factors are considered: (1) the scope and content of the prior art; (2) the differences between the art and the claims at issue; (3) the level of ordinary skill in the art at the time the invention of the asserted patent was made; and (4) objective evidence of nonobviousness. In order to determine whether claim elements are found in the prior art, it is necessary to compare the properly construed claim language of the patent with the teachings of the prior art.

27. I understand that certain factors—often called "secondary considerations"—may support or rebut an assertion of obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the alleged invention, skepticism of those having ordinary skill in the art at the time of the alleged invention, unexpected results of the alleged invention, any long-felt but unsolved need in the art that was satisfied by the alleged

invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field. I further understand that there must be a nexus—a connection—between any such secondary considerations and the alleged invention.

28. I further understand that a claim can be found obvious if it unites old elements with no change to their respective functions or alters prior art by mere substitution of one element for another known in the field, with that combination yielding predictable results. While it may be helpful to identify a reason for this combination, common sense should guide, and there is no rigid requirement for a teaching, suggestion, or motivation to combine. When a product is available, design incentives and other market forces can prompt variations of it, either in the same field or different one. If a person having ordinary skill in the relevant art can implement a predictable variation, obviousness likely bars patentability. Similarly, if a technique has been used to improve one device, and a person having ordinary skill in the art would recognize that the technique would improve similar devices in the same way, use of the technique is obvious.

29. I also understand the following rationale may support a finding of obviousness:

- Combining prior art elements according to known methods to yield predictable results;

- Simple substitution of one known element for another to obtain predictable results;
- User of known technique to improve similar devices (methods, or products) in the same way;
- Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- "Obvious to try" — choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- Known work in one field of endeavor may prompt variations of it for use in either the same field or a different on based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art;
- Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

30. I also understand that it is impermissible to use the claimed invention itself as a blueprint for piecing together elements in the art. In other words, it is impermissible to use hindsight reconstruction to pick and choose among disclosures in the prior art to reconstruct the claimed invention.

IV. DESCRIPTION OF THE TECHNOLOGY

A. Internet Communications in General

31. The Internet allows devices, such as servers, smartphones, and tablets, to communicate by exchanging digital information in the form of packets across a global system of interconnected computer networks. To exchange information reliably and effectively, each device and network must use a standard set of rules,

messages, and messaging formats, known as communication "protocols." Ex.1024, 23, 417.

32. The term "Internet" is a contraction of the phrase "interconnected network". The Internet is not one network, but is instead many interconnected networks. It consists of many groups of networks, including the following examples, interconnected at network access points.

- Backbones (*e.g.*, AT&T, Verizon, and Sprint)
- Regional Networks
- Local Networks (*e.g.*, Cornell computer network)

33. The first piece of the Internet was *ARPANET*. It was built by DARPA in the late 1960's to interconnect research centers, military bases, and government locations of various types. The Defense Communication Agency (DCA) converted ARPANET into a research tool in 1975. ARPANET was split into MILNET (for military bases) and ARPANET (for research centers) in 1983.

34. One of the primary foci of the ARPANET research was the development of interconnection protocols. The first milestone in this area was the Network Control Program (NCP - 1969). As a monolithic program, it exhibited scalability problems as the Internet evolved. In the 1970s, NCP was broken up into TCP (the Transmission Control Protocol) and IP (the Internet Protocol). This was the beginning of the suite of protocols that we now referred to as TCP/IP.

35. In 1983 DARPA insisted that all computers wanting to connect to ARPANET use TCP/IP. This was a critical step, as it allowed for different computer technologies to share the Internet.

36. ARPANET was eventually replaced by NSFNET, in part to reflect the increasing prominence of academic research. In 1986, the National Science Foundation (NSF) built a backbone network to interconnect four NSF-funded regional supercomputer centers and the National Center for Atmospheric Research (NCAR). The original architecture consisted of

- backbone
- mid-level networks (regional, discipline-based, and supercomputer consortia)
- campus networks

37. It is important to note that NSFNET traffic was limited to non-commercial use.

38. The first backbone used 56 Kbps leased lines. The second backbone, developed by NSF, Merit, IBM, and MCI and installed in 1988, used 448 Kbps leased circuits to interconnect 13 nodal switching centers. The third backbone (1989) began using T1 lines (primarily fiber optic), and has since migrated to T3 (45 Mbps - 1992).

39. The migration to gigabit speeds was fueled in part by a surge in commercial activities that began in April 1995, when the U.S. Government stopped its funding of NSFNET and allowed the presence of commercial activity.

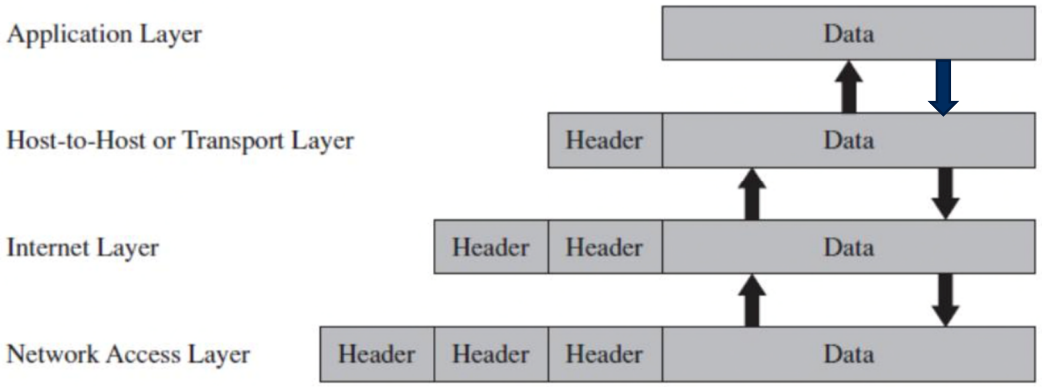
40. The Internet communication technologies used today are based on protocols developed decades ago including the transmission control protocol (TCP) and internet protocol (IP), the more prominent members of the "TCP/IP" protocol suite (a suite that actually includes dozens of protocols). *See* Ex.1025; Ex.1024, 37.

41. TCP/IP aimed to ensure reliable, low-latency, end-to-end data transmission between network-connected devices by dividing transmissions into "packets" and specifying rules for transmitting, routing, and receiving packetized messages. *See* Ex.1024, 39, 261. Latency, as it relates to live streaming, is the time delay between when an event occurs and when the distributed content related to that event, e.g., video, audio, or other data, is displayed on a user device. *See, e.g.*, Ex.1003, 2:31-40 ("'[L]atency,' *i.e.*, the delay between a first user generating a live video stream for transmission via the Internet and a second user receiving a copy of the live video stream via the Internet for viewing.").

42. Implementation of TCP/IP communications in operating system software often relies on an application programming interface (API). *See* Ex.1026, 5. APIs expose functions that allow applications to exchange information over the Internet while handling the low-level details of encapsulation, formatting,

transmitting, and receiving TCP segments and IP packets. By leveraging APIs, developers can build networked applications that operate on underlying network infrastructure without the need for a sophisticated understanding of lower-level implementation details, which are managed by the "protocol stack" within the operating system. *See* Ex.1024, 377.

43. Generally speaking, packets from higher layer protocols are placed in the information fields of the next lower-layer packets. This process, called "encapsulation," is depicted below.



44. To access this stack structure, APIs for TCP/IP communication generally rely on abstractions such as "ports" and "sockets". Ex.1024, 378. A port is a number that is assigned to uniquely identify a connection endpoint and associate the connection with a specific process or service in the host software. *Id*. For example, port 80 is uniformly used to represent Hypertext Transfer Protocol (HTTP) connections used on the World Wide Web to serve webpages. *See id*. Sockets are

software structures that represent the connection for sending and receiving data across the network. *Id.*

45. An application designed to exchange data over the Internet (*e.g.*, a web browser) will first open a socket with a specific IP address and transport layer port in order to establish a connection with a remote endpoint. *See id.*, 378-379. The application may then read and write to the opened socket using API functions that implement the low-level details of the TCP/IP protocol required to exchange data over the network. *See id.*, 333-335. APIs for establishing and communicating using sockets were developed as early as 1983 with the "Berkeley sockets" API, part of version 4.2 of the BSD Unix operating system. Modern operating systems continue to implement similar APIs based on the Berkeley socket interface.

B. Network Communication Models

46. Communications involving multiple hosts, each with multiple ports (and thus multiple sockets) may take on various topologies, which characterize the relationships between the hosts and/or ports involved in a connection. Ex.1024, 12-15. Possible connection topologies include one-to-one, one-to-many, and many-to-many. *See id.*, 12-15, 270. A one-to-one connection topology exists when only one source host and one destination host are involved. There, the communication is between one source socket (representing the source host's IP address and port number) and one destination socket (representing the destination host's IP address

and port number). *Id.*, 392-395. When multiple destination hosts wish to receive information from the same source host, the connection topology may be one-to-many or many-to-many. *See id.*, 12-15.

47. In a one-to-many topology, the communication may be from the same source socket at the source host to multiple destination sockets (one at each destination host). *See id.*, 1024, 12-15. A web server is a common example, where all users connect to the same socket (using port 80). *See id.*, 378, 435. Content Distribution Networks (CDNs) are another prominent example in which many destinations may be observing the same video event, such as the Olympics. Ex.1023. Alternatively, the network topology may be many-to-many, where multiple source sockets are available to transmit information to multiple destination sockets. *See id.*, 12-15, 242. In this topology, each destination socket may be assigned its own unique source socket, or multiple destination sockets may share a source socket, with different sets of destination sockets sharing different source sockets. *See id.*, 270.

48. Various protocols have been developed to accomplish one-to-many and many-to-many communications over a network. Examples include multicast addressing using the User Datagram Protocol (UDP) and "IP Multicast," which is an extension to the standard IP network-level protocol that enables sources to send a single copy of data to multiple devices that explicitly request the information. Ex.1024, 347, 350, 355. Once an IP Multicast user requests access to a specific

multicast, the multicast messages pass to the TCP/IP protocol stack, making them available as input to the user's application (*e.g.*, a video viewer). *Id.*, 7.

C. Protocols for Real-time or Low-latency Data Transmission

49. In addition to the TCP/IP protocols described above, before the '687, and '697, developers and standard development organizers had created additional protocols to facilitate real-time transport of video and other data over the Internet.

1. Real-Time Transport Protocol (RTP)

50. RTP is a communication protocol originally developed in 1996 that optimizes live audio and video data transmission. *See* Ex.1027; Ex.1028. RTP is commonly used in protocols requiring real-time audio and video data streaming over the Internet, such as the Voice Over Internet Protocol (VoIP). *See* Ex.1024, 400, 402.

2. HTTP Polling

51. HTTP is a protocol that is built on top of TCP/IP. Ex.1024, 392. HTTPS is a secure version of HTTP that uses encryption and verification to protect data. HTTP polling consists of a sequence of request-response messages between a client and a server. Upon receiving a request, the server responds with a new message, if there is one, or with an empty response if no new message is available for that client. After a short time Δ, called the polling interval, the client polls the server again to see if any new messages are available. *See* Ex.1024, 45-46; Ex.1047.

Various real-time applications including chat, online games, and text messaging used HTTP polling in the 2016 timeframe.

3. WebSocket

52. WebSocket is a client-server communication protocol standardized by the Internet Engineering Task Force in 2011 that is bi-directional, full-duplex, and continuous. *See* Ex.1029; Ex.1030; Ex.1043. Real-time web applications commonly feature WebSocket connections to facilitate a persistent stream of data from a backend server to the end client. Ex.1030; Ex.1018 ¶6. Once a WebSocket connection is established between a server and a client, the client receives instant updates from the server without the need to poll. Ex.1030.

53. In any content distribution system, the distribution server needs to transmit data to receiving client devices. If the system uses sockets to communicate, by way of analogy, a transmit server is like a restaurant, and the client devices are all homes, and each buildings' door is a socket. *See* Ex.1022, 87. Food is delivered out of the restaurant's door, *i.e.*, socket, and accepted through each separate homes' door, *i.e.*, individual sockets. *Id.* The roads between the restaurant and homes are the internet's TCP protocol. Moreover, each home can send someone or something to the restaurant--it is not a one-way communication system. *Id.*, 160. This connection is typically established when a client device sends a request to a running server process (live streaming content, i.e., open restaurant) to "initiate a TCP

connection to the server" (knocks on the door). *Id*., 160-61. To do so, the client device must be able to specify "the address of the server process, namely, the IP address of the server host and the port number of the server process." *Id.* Afterwards, the connection can be established at the socket with the identified IP address and port number. Thus, the same server process accepts connections to multiple destination sockets (client devices) which creates a one-to-many topography.

4. Stream Control Transmission Protocol (SCTP)

54. SCTP is an IP transport layer protocol adopted in 2001 that provides reliable, in-sequence data transport. *See* Ex.1031; Ex.1032. As a message-oriented communication protocol, SCTP is ideally suited for applications requiring multiple, independent streams of messages to be delivered in parallel—in other words, "multi-streamed"—because the protocol does not require a dedicated socket for each data stream. *See* Ex.1033. SCTP's multi-streaming capability is particularly useful for controlling congestion in multimedia streaming over the Internet. Ex.1034.

5. Real-Time Messaging Protocol (RTMP)

55. RTMP is a low-latency data transmission protocol in use as of at least 2012 that is designed to support live online streaming in social media platforms, streaming platforms, and media servers. *See* Ex.1035, 3; Ex.1036, 1. RTMP data transmission begins with a client-initiated handshake between the client and server.

Ex.1036, 7. Once a connection is established between the client and server, RTMP operates as a bi-directional multiplexing service over TCP to stream live and on-demand video or audio data from the server to the client device. *See* Ex.1035, 3. The Challenged Patents use this traditional streaming protocol:

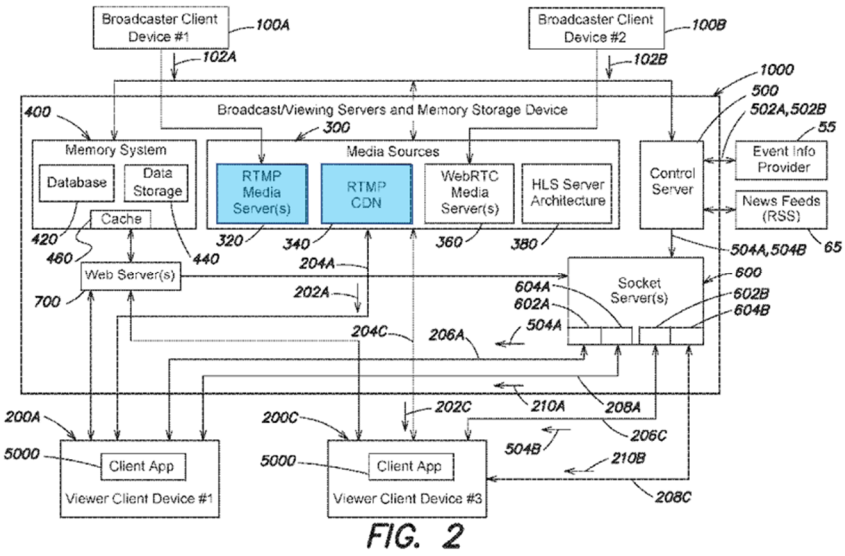


FIG. 2

See, e.g., Ex.1003, Fig. 2.

6. Web Real-Time Communication (WebRTC)

56. WebRTC is a technology first implemented in 2011 that enables real-time audio and video streaming within web browsers and mobile applications without an intermediary. Ex.1037, 1; Ex.1038, 3. WebRTC facilitates peer-to-peer communications between mobile device web browsers by creating an RTCPeerConection, a signal through which mobile devices exchange hardware and codec compatibility information. Ex.1037. Once a connection between devices is created, WebRTC offers video calling, video chat, and file-sharing capability with

minimal plugins. Ex.1038, 3. The Challenged Patents also use this traditional streaming protocol:

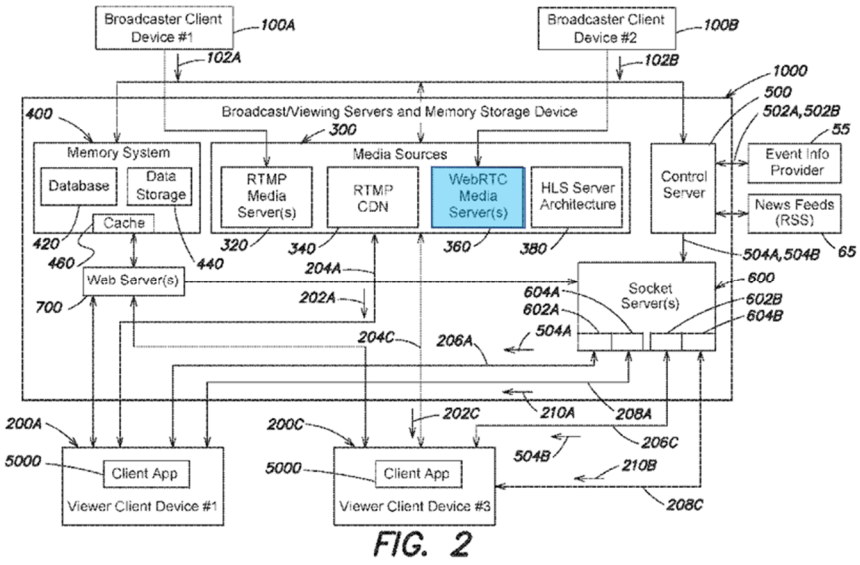


FIG. 2

See, e.g., Ex.1003, Fig. 2.

7. Dynamic Adaptive Streaming Over HTTP ("DASH" or "MPEG-DASH") / HTTP Live Streaming (HLS)

57. MPEG-DASH is an adaptive streaming protocol adopted in 2011 that enables segmented multimedia content delivery at different bit rates. Ex.1039, Abstract; Ex.1040, 2. In the MPEG-DASH process, an origin server pushes encoded, segmented stream data to the device over the Internet, as well as manifest files identifying alternative streams available to the player and their corresponding URLs. *See* Ex.1040, 2. Once the user chooses a stream of the desired bitrate, a web server transmits the data in the selected bit-rate stream to produce smooth playback on the user's device. *See* Ex.1041, 1; Ex.1035, 4. HLS, another adaptive streaming protocol adopted in 2009, functions similarly and can be used in many of the same

applications as MPEG-DASH. Ex.1041, 1. The Challenged Patents also use this traditional streaming protocol:

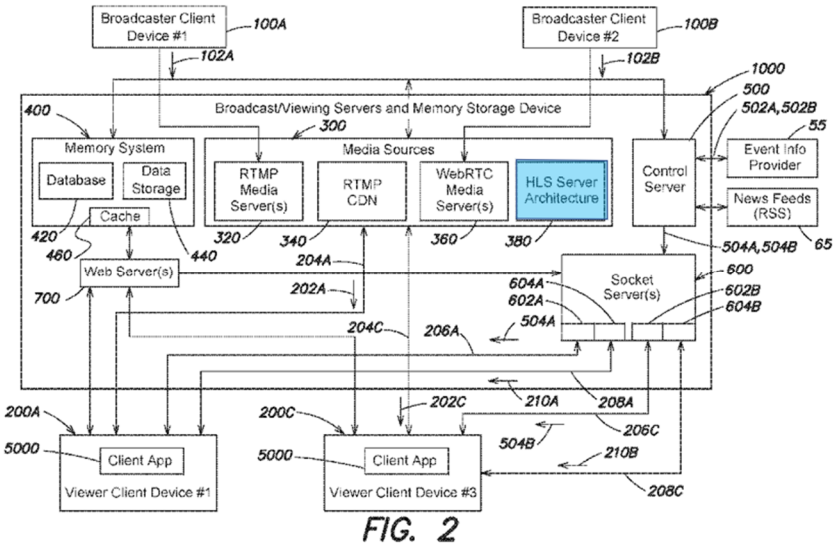


FIG. 2

See, e.g., Ex.1003, Fig. 2.

D. Internet Live Video Streaming

58. After the World Wide Web became widely available to the public in 1993, the Internet quickly evolved from a predominantly text-based medium to one that facilitated multimedia content delivery. *See* Ex.1024, 16-17; Ex.1042. In the late 2000s, video streaming services also developed technologies for dynamically combining video streams with relevant text-based data. For example, in 2007, YouTube implemented an "overlay" system that superimposed advertising content relevant to the video onto a video stream without interrupting playback. Ex.1044, 1.

59. Not long after, numerous services developed tools for streaming live (rather than prerecorded) media over the Internet to multiple users in real time. Such examples available to the public before the Challenged Patents were filed included Livestream (founded in 2007), Twitch (same), Wowza (same), TikTok (formerly Musical.ly, 2014), Periscope (2015), and Facebook Live (2016). *See* Ex.1003, 2:4-27.

V. OVERVIEW OF THE CHALLENGED PATENTS

60. The Challenged Patents share a common specification, and the claims substantially overlap. Their family relationship is shown below:



61. Because all of the Challenged Patents share a common specification, for convenience, I have cited to the '687 as representative.

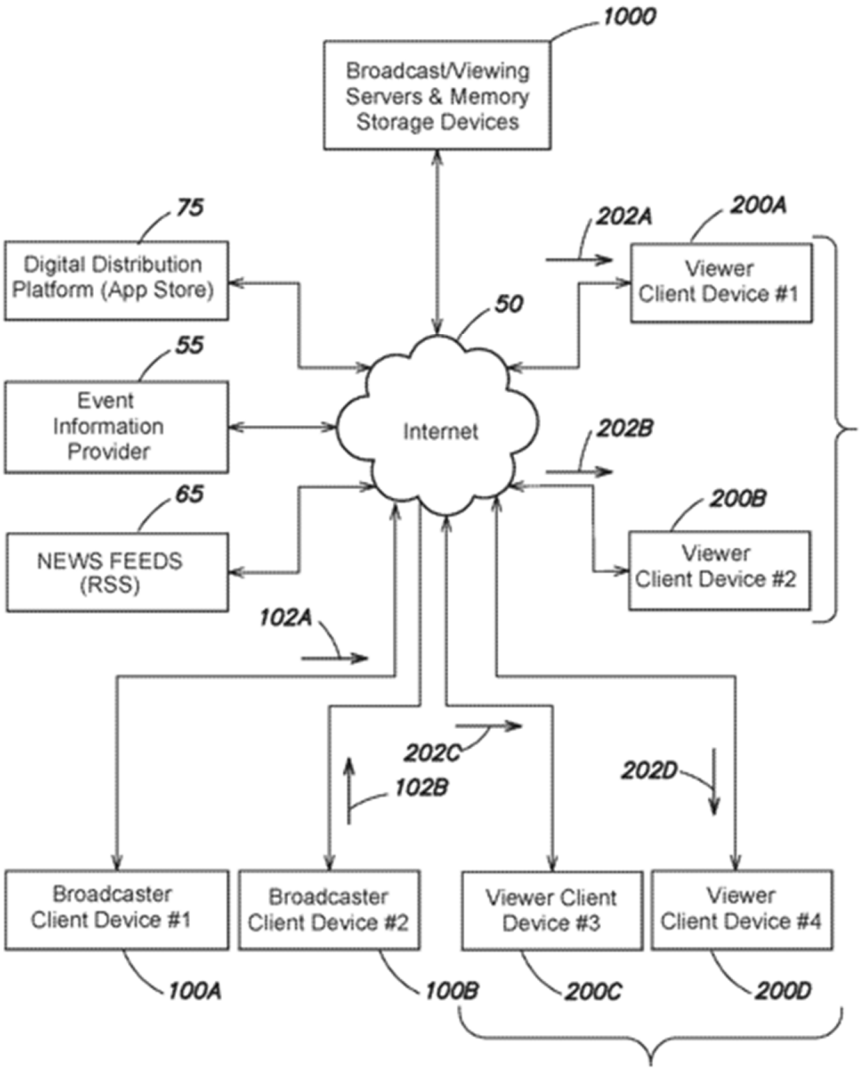
A. Alleged Problem in the Art

62. According to the '687, there was a need to improve "conventional techniques for transmission of digital content via the Internet" that included a "latency" problem. Ex.1003, 2:31-40. Latency is "the delay between a first user generating a live video stream for transmission via the Internet and a second user

receiving a copy of the live video stream via the Internet for viewing." *Id*. Specifically, "live video streams with relatively high quality (e.g., high definition HD and high bit rate, such as 2 to 5 megabits per second)," may have difficulties "synchronizing such … copies of a live video stream" amongst a large number of viewers. *Id.*, 2:43-56. The '687 purports to solve this problem, ensuring a "low-latency viewing experience for significant numbers of viewers" and "appropriate synchronization of event information across all broadcasters of an event and their associated viewers." *Id.*, 5:28-37.

B. Alleged Invention of the Challenged Patents

63. In my opinion, 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." *Id.*, 3:7-20; *see also id.*, 7:16-37. A block diagram of the system is shown in Figure 1A:



Id., 11:19-25, Fig. 1A. 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.

64. Figure 2 shows additional system details. For instance, Figure 2 depicts viewer client devices ("VCDs"), media sources, web servers, a control server, a socket server, event sockets, an event information provider, and a memory system.

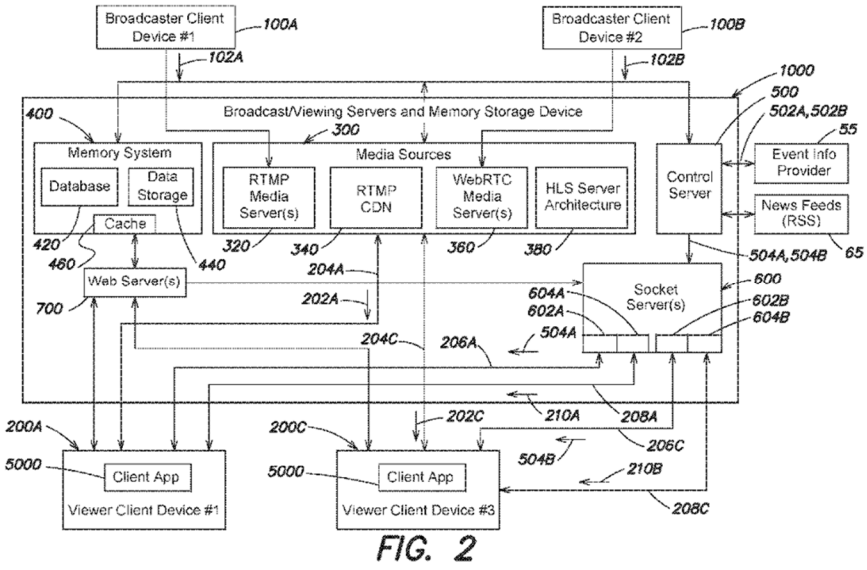


FIG. 2

Id., Fig. 2. I address how the '687 describes certain components relevant to the claims below.

65. The '687 describes media sources as generic "computer servers" comprising known components such as "processors, memory, and … communication interfaces." *Id.*, 19:17-23, 55:15-25. Although in some implementations the media sources may include media servers directed to specific known protocols, such as RTMP and HLS, the media sources' purpose is to "receive a live stream of video-based commentary from a given broadcaster client device and provide copies of the live stream of video-based commentary to one or more [VCDs]." *Id.*, 19:17-31.

66. The ’687 describes a control server as generic server equipment comprising one or more processors, memory devices, and communication interfaces, *id.*, 55:15-25, that retrieves “event information” from an “event information provider” to pass to the socket server, discussed below. *Id.* 19:51-55, 21:46-58. The event information is “germane to” a particular “live sporting event,” *e.g.*, the event’s score information. *Id.* 19:37-56.

67. The ’687 describes a web server as a generic server designed to facilitate communications “between the servers/memory storage devices [] and one or more mobile broadcaster/viewer client device executing the client app.” *Id.* 18:55-65. According to the ’687, “much of the instructive communication between the client devices and the servers/memory storage devices [] occurs via the web server(s).” *Id.* 18:64-67. The webserver also facilitates communications “to and from web-based client devices (that access the web server(s) via a web portal at a particular URL).” *Id.* 18:55-65. The web server accomplishes this, in one implementation, by “select[ing] a particular media server of the media sources [] to which a given broadcaster connects to provide the broadcaster’s live stream of digital content ... .” *Id.* 19:37-47.

68. The ’687 describes socket server(s) as generic servers. *Id.* 55:15-25. The socket server’s purpose is to “facilitate communication” of “synchronized event information retrieved by the control server” between the control server and the

VCDs. *Id.* 20:7-15. The socket server uses "one or more sockets … dedicated to the particular event" allowing VCDs "to establish an event information channel with the socket server." *Id.* 20:15-21. The socket server then shares event information in "event messages" via the event information channel "in a synchronized manner" to all viewers "following the particular event." *Id*. The socket server may facilitate communication of "chat messages" in a similar manner. *Id.* 20:22-54.

69. The '687 explains that the "connections between a given client device and a particular socket of a socket server may be persistent authenticated connections (e.g., with IP-based fingerprint identifiers for anonymous users)" and that "event messages" among other data carried on a channel between a client and a socket are "data packets including," *e.g.*, "event information." *Id.* 20:55-58, 20:64-21:4. Each event socket has a corresponding "event identifier" (either an "EventID" or "StreamID") that the VCD "uses ... to connect to [the] socket," *e.g.*, by using a "URL including the ... EventID in a path of the URL." *Id.* 21:63-22:8. Like known servers before the alleged invention of the '687, messages carried on channels between sockets of the "socket server" use traditional IP mechanisms, *e.g.*, data packets, and each "socket" is uniquely addressable, *e.g.*, by using an identifier that may be contained, *e.g.*, in a URL. *Id.* 20:55-21:4, 21:63-22:16.

70. According to the '687, a "'communication channel' refers to a physical or logical connection over a transmission medium to convey information signals

from one or more senders to one or more receivers," which the '687 notes "would be readily appreciated in the relevant arts." Ex.1003, 5:38-45. As examples of Internet communication channels, the '687 describes generic Internet communication channels between a sender and receiver: (1) a "video" channel between a media server and a VCD to convey video-based commentary, (2) an "event information" channel between a socket and the client device to convey event information, and (3) a "chat/system event" channel between another socket and the client device to convey chat and other information. *Id.* 5:45-58.

71. Although the '687 claims that the foregoing server architecture and its functionality are "inventive," as explained in more detail herein, there is a reasonable likelihood that the prior art discloses or renders obvious every element of at least one the Challenged Claims.

VI. LEVEL OF ORDINARY SKILL IN THE ART

72. I understand that a POSA is a hypothetical person presumed to have the skill and experience of an ordinary worker in the field at the time of the alleged invention.

73. I understand that there are multiple factors relevant to determining the level of ordinary skill in the pertinent art at the time of the alleged invention. These factors include the education of active workers in the field, the sophistication of the technology, the type of problems encountered in the art, and the prior art solutions

to those problems. I further understand that the POSA can embody the knowledge and experience of multiple individuals with different educations and backgrounds.

74. In determining the characteristics of a hypothetical POSA of the Challenged Patents at the time of the claimed invention, I considered several things, including the type of problems encountered in this field, and the rapidity with which innovations were made. I also considered the sophistication of the technology involved, and the educational background and experience of those actively working in the field, and the level of education that would be necessary to understand the Challenged Patents.

75. Finally, I placed myself back in the relevant period of time and considered the state of the art and the level of skill of the engineers working in this field at that time.

76. In my opinion, 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. Additional education or experience in related fields could compensate for deficits in the above qualifications. By this definition, as of August 2016. I exceeded these qualifications.

VII. CLAIM CONSTRUCTION

A. The ’687 and ’697 Patents

77. For purposes of this declaration, I have been asked to apply the plain and ordinary meaning of the claim terms as they would have been understood by a person of skill in the art (“POSA”). I reserve the right to respond to any specific construction proposed by Patent Owner. However, my ultimate opinions of invalidity of the Challenged Claims would not likely change under any reasonable construction of the claims, to the extent that the claims can be understood by a POSA.

78. I have also been advised that, to the extent that the Challenged Claims are system claims, many of the elements recited in the claims constitute nothing more than recitations of intended use and/or functionality and may not serve to further limit the claimed system. Similarly, many of the claim elements refer to the particular content or types of data being received or transmitted (*e.g.*, “event video-based commentary,” “event information,” “live sporting event,” etc.), which I also understand cannot limit the scope of the Challenged Claims. Thus, I expressly reserve the right to respond to any assertion regarding the limiting effect of any such claim element.

VIII. SUMMARY OF THE PRIOR ART

A. U.S. Patent Pub. No. 2015/0163379 ("Herzog")

79. Herzog is a U.S. Patent Publication that was published on June 11, 2015. Herzog is generally directed to a content delivery system and network architecture that delivers live video and audio streams of sporting events to users while also providing meta-data associated with the sporting event to a user over a separate feed, where the video/audio stream and meta-data are synchronized. *See* Ex.1015, Abstract, ¶¶3, 18.

80. Herzog states that "a need exists for improved technologies for synchronizing information (for example, HLS video and meta-data) that is published and processed at different rates." *Id.* ¶4. Herzog explains that it "may be desirable to provide a mobile device, such as a tablet" with a "video that is streamed live from the Internet to a client playing using HTTP Live Streaming (also known as HLS)" while also providing "additional information related to the HLS video," such as meta-data. *Id.* ¶3.

81. Herzog uses an "Indianapolis Car Race" as a primary example illustrating how its system transmits live "video, audio, and metadata … from a sports venue via different communication paths," to a client device, but notes its system can be applied to any live sporting event. *Id.* ¶¶16, 83. Herzog's Figure 4, shown below, depicts an illustrative streaming network:

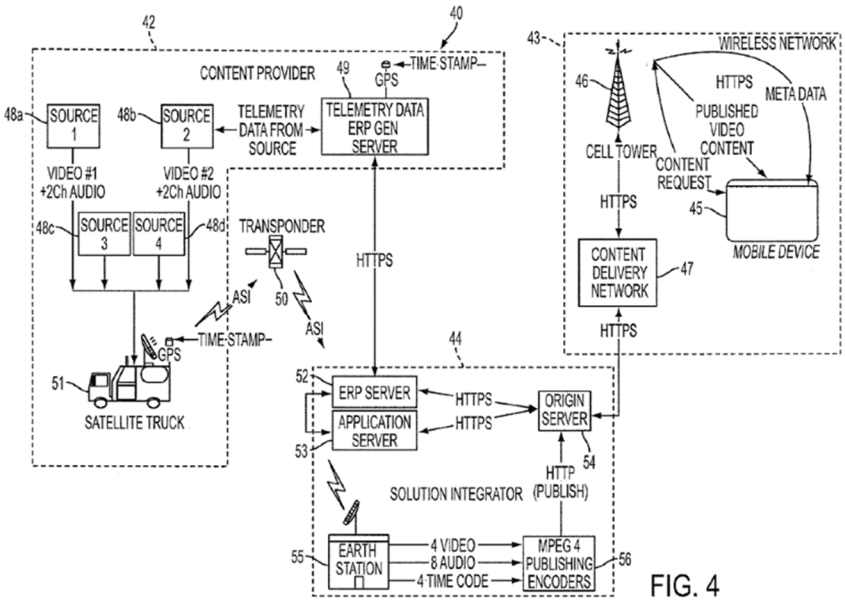


FIG. 4

Id. Fig. 4.

82. Herzog's Figure 4 shows a content provider, a solutions integrator, and a wireless network. *Id.* ¶¶43, 45. The content provider "is configured to provide video and audio data of an event occurring in real time and provide telemetry data (meta-data) including several parameters changing during the event." *Id.* ¶45. The meta-data "provides additional information related to the HLS video" and in the context of a car race, such as the example shown in Figure 4, "may include varying parameters regarding the status of each car and driver in the race" including "lap number, car speed and acceleration, tire pressure, etc." *Id.* ¶¶3, 45. Video and audio data includes "multiple video/audio streams of race cars and drivers, as they are imaged at different locations on the race track." *Id.*

83. Figure 4 shows "four video/audio streams provided from four different imaging sources, namely sources 48a-48d." *Id.* ¶46. "As shown, each source provides one video stream and two channels of audio …." *Id.* These channels are transmitted to an Enhanced Result Protocol (ERP) generator server configured to provide telemetry data in HTTPS format to the solutions integrator. *Id.* ¶47. The solutions integrator is depicted in another embodiment below in Figure 6:

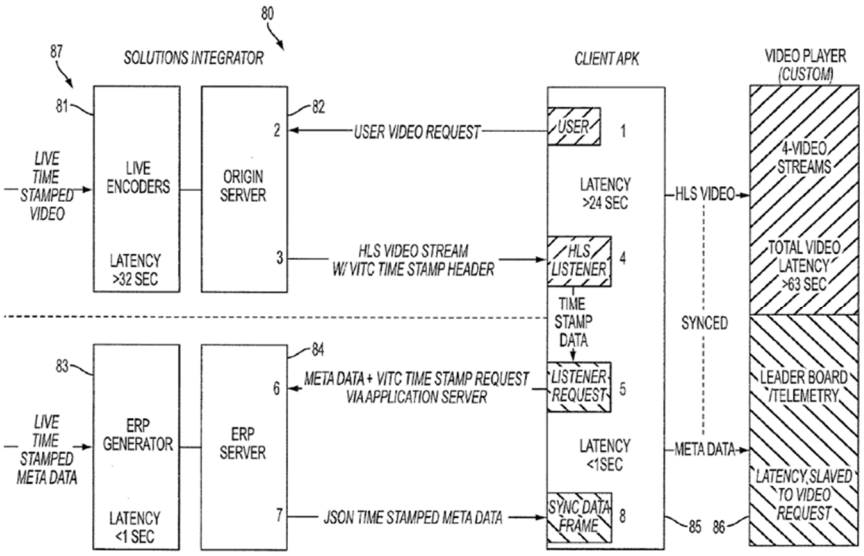


FIG. 6

Id. Fig. 6.

84. Once live encoders receive "time stamped video/audio data and convert the same into an MPEG stream with a VITC time stamp header," they then send the data to an origin server to "publish[] the data as an HLS video stream with a VITC time stamp header." *Id.* ¶67. In parallel, the ERP generator extracts various parameters from the live time stamped meta-data, then sends those parameters to the

ERP server for publishing "in an HTTPS communications format," such as a .JSON time stamp. *Id.* ¶¶67-68. At this point, the data is ready to be streamed to a client device upon request by a client application package (APK).

85. To begin streaming, a client device "initiates a video request of one or more channels of video/audio for viewing a live event on the device display." *Id.* ¶70. The origin server receives the request and begins "streaming HLS video of the live event with the VITC time stamp header to the client application." *Id.* ¶¶71-72. Upon viewing a video of the live event, a user can "initiate[] a request" to the ERP server "for meta-data that is associated with the VITC time stamp of a particular video channel on the display of the video player." *Id.* ¶¶74-75. In response, the ERP server transmits synchronized, .JSON-time-stamped meta-data to the client application for display on the video player. *Id.* ¶77. As shown above in wireless network 43, the meta-data is then provided through the origin server to the user device "through one or more networks." *Id.* ¶¶3, 27, 51. At the user device, "all the data may be displayed … in correct time sync" because the "video and audio data and the meta-data are both time stamped with consistent (e.g., GPS) timing protocols." *Id.* ¶18.

86. Herzog also discloses that a general server can be substituted for several servers used to execute a specific function. *Id.* ¶66 (describing a "solutions

integrator" as "more generally, a server capable of receiving two types of signals, namely RF video/audio signals and HTTPS signals ... .").

B. U.S. Patent Pub. No. 2014/0229992 ("Ellis")

87. Ellis is a U.S. Patent Publication that was published on August 14, 2014. Ex.1016, Cover. Ellis is generally directed to systems and methods that enhance "sports-related media content access and display." *Id.*, Abstract. Ellis discloses an interactive system for watching sports content, including "wagering and fantasy league support." *Id.*

88. Ellis identified the need for current delivery systems to provide "a custom set of features and functionality for specific types of media content, such as sporting events …." *Id.* ¶4. Ellis explained that the current systems "fail to take full advantage of today's high-end user equipment devices." *Id.* ¶5. To solve the above problems, Ellis discloses broadcasting systems and methods for "enhanced sports-related media content access and display." *Id.*, Abstract. Ellis's Figure 1 is an illustrative media system:

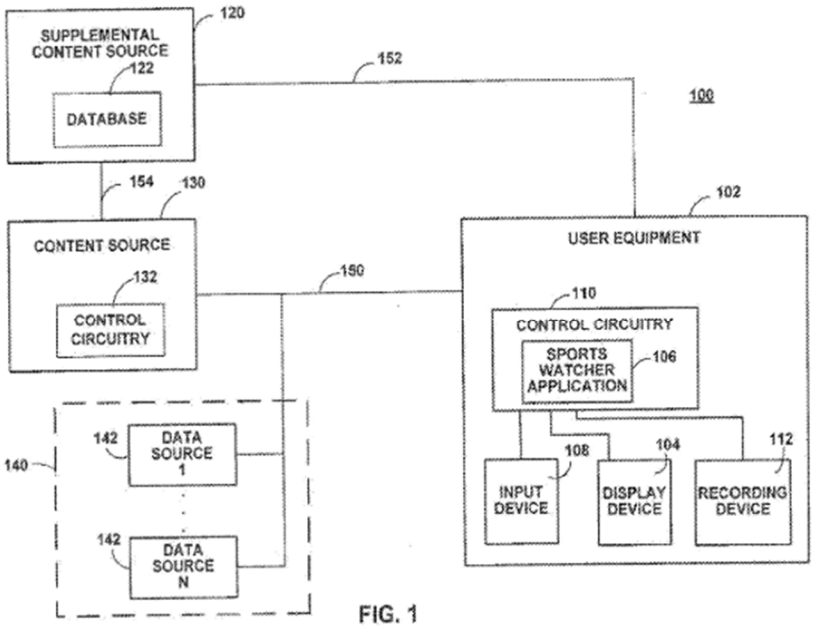


FIG. 1

Id. Fig. 1.

89. As depicted, Ellis's system contains a content source, a supplemental content source ("SCS"), data sources, user equipment, and a sports watcher application. *Id.*, Fig. 1. Even though the content source, SCS, and data sources "are shown in Fig. 1 as separate elements," their "functionality may be combined into a single mechanism" provided "from a single system at a single facility" or "their functionality may be provided by multiple systems at multiple facilities." *Id.* ¶47.

90. Ellis describes how users connect to available video feeds by selecting a live event corresponding to a "source identifier (e.g., channel identifier)," as shown in Figure 2B:

220

222 Tuner	224 Primary Source ID
1	<sourceid_2>
2	<sourceid_1>
3	<sourceid_8>
4	<sourceid_4>

230

232 Source ID	234 Type	236 Profile Score
<sourceid_1>	Game	90
<sourceid_2>	Game	95
<sourceid_3>	Commentary	45
<sourceid_4>	News	75
<sourceid_5>	Movie	25
<sourceid_6>	Movie	35
<sourceid_7>	Highlights	65
<sourceid_8>	Game	80

Id., Fig. 2B.

91. Then, the content source distributes the live-event streams to multiple user devices over separate communication paths 150. *Id.* ¶¶39-40, Fig. 1. The content source can be "any suitable source of media content, such as, a[n] … Internet or network media/webserver, or any other suitable ... system for originating or distributing passive or interactive content to user equipment 102." *Id.* ¶40. User equipment includes "any equipment suitable for presenting media content to a user" including "computer equipment." *Id.* ¶48. Communication path 150 may be a satellite, fiber-optic, cable, or any suitable wired or wireless path. *Id.* ¶45. For example, Ellis discloses IPTV, where the user may receive the content "on a particular network address (e.g., multicast address) on which an IP socket is currently joined." *Id.* ¶74.

92. While the system provides the video feed over communication path 150, the SCS further provides real-time information over a separate communication path 152. *Id.* ¶¶41-42, Fig. 1. The SCS can be "any suitable content source

configured to provide real-time sports scores and game summary information, … spread and wagering data, and any other sports-related information to user equipment 102." *Id.* ¶41. Communication path 152 may be a satellite, fiber-optic, cable, or any suitable wired or wireless path. *Id.* ¶45. Ellis explains that "in some embodiments, this sports-related information is provided to user equipment 102 as a plurality of real-time data feeds" where each "data feed may be individually accessed or subscribed to …." *Id.* ¶41.

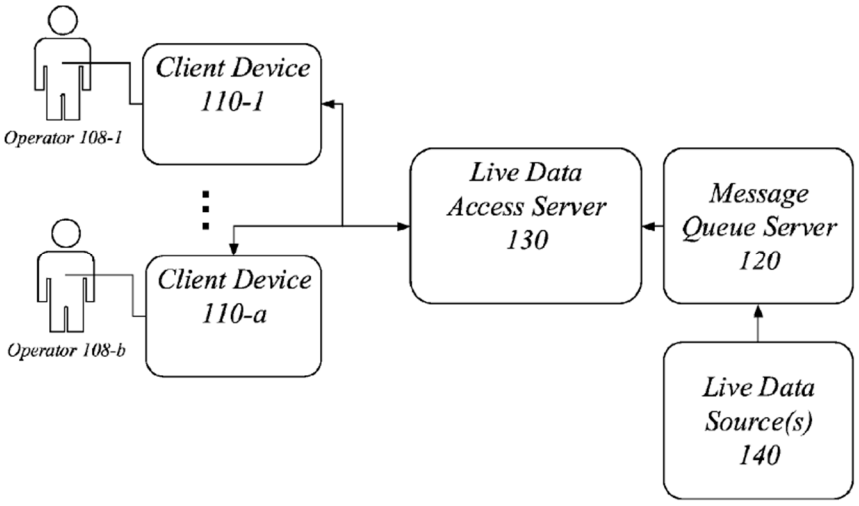
C. U.S. Patent Pub. No. 2016/0036910 ("Spivey")

93. Spivey is a U.S. Patent Publication that published on February 4, 2016. Ex.1017, Cover. Spivey is directed to a live event data delivery system that can "broadcast live event data in real-time," including live data of ongoing sporting events. *Id.*, Abstract, ¶27.

94. Spivey identifies the problem of increased demand for "live event data in real time." *Id.* ¶2. As demand increased, "content providers may encounter obstacles to providing the data …, in real-time, and with minimal delays." *Id.* Conventional techniques "may cause latency or a delay from the time the data is received by the live event data content provider to when the client receives the data." *Id.,* ¶28.

95. Spivey addressed the above problem and disclosed an apparatus and methods that "increase the speed at which live event data (e.g. sports scores and

breaking news), are broadcast … using topic queues and open duplex communication techniques, among other innovative techniques." *Id.* ¶27. Spivey reduced real-time data delay from over 30 seconds to "an order of milliseconds." *Id.* ¶28. Spivey Figure 1 "illustrates an embodiment of a system for broadcasting live event data:"



Id., Fig. 1.

96. Figure 1 depicts live data source(s), a message queue server, a live data access server, and client devices (110-1…110-a). *Id.*, Fig. 1. Client devices (110-a) "may include, without element, … a computer, … digital television, or any combination thereof." *Id.* ¶¶35, 39 ("Client device 110 may include any electronic device as described above."). The message queue server "may comprise one or more electronic devices capable of receiving, processing, and sending information for the system 100." *Id.* ¶ 35 (detailing possible examples including "a server, a server array

or server farm, a web server, a network server, [and] an Internet server" among others).

97. The live data source(s) "may include data feeds, generated by computing devices associated with a live event …." *Id.* ¶37. Also, "human observers of a live event" may gather and enter data into "an interface that communicates the gathered data to message queue server 120 as live data." *Id.*

98. In certain embodiments, the message queue server 120 communicates directly with client devices (110-1…110-a) without the use of a live data access server. *Id.* ¶¶34, 38, 46, 53, 84. These embodiments, which divide the message queue server into at least two components, 450 and 410, is shown as operating environment 400, in Figure 4:

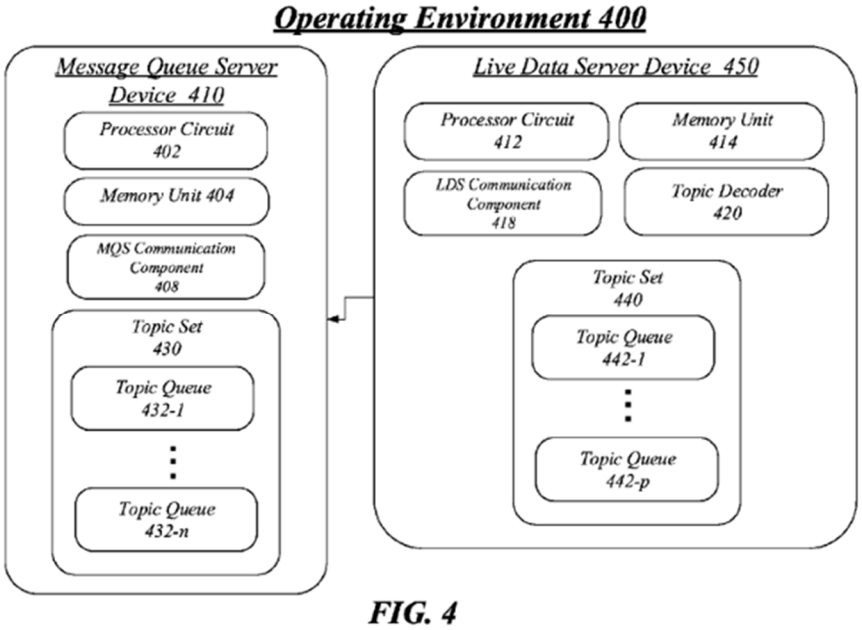


FIG. 4

Id., ¶53, Fig. 4.

99. In order to provide real-time data with reduced latency, LDS communication component receives "live data … payloads." *Id.* ¶55. The payloads include score information as well as "a player injury, an error, a record, a substitution, [or] statistic[s] …." *Id.*, Claim 11, Fig. 10. Live data server device ("LDSD") sorts the data, using attached topic labels. *Id.* ¶¶47, 49. A topic label could be a text string, "gamecenter.nfl.ravenssteelers," indicating to the live data server to place the attached data payload in the topic queue for the Ravens/Steelers game. *Id.* ¶66. The LDSD then provides the data to the message queue server device ("MQSD"), which stores the data into the appropriate topic queues. *Id.* ¶¶47, 54-56.

100. Spivey describes topic queues as an event information storage location that "may be a queue data structure, … where data payloads are placed …." *Id.* ¶51. Topic queues are locations that "may be hierarchically related." *Id.* ¶52. For example, "'gamecenter.nfl' may apply to all data about the National Football League" and a dependent topic may be "'gamecenter.nfl.ravenssteelers' which may relate to a specific game between two NFL teams." *Id.* "A third topic may be "gamecenter.nfl.ravenssteelers.stats" which may relate to statistics about the specific game." *Id.* ¶52.

101. Multiple topic queues for multiple different types of data (*e.g.*, statistics and plays) can exist for the same live event. *Id.* ¶¶52, 104 ("If there were a fourth topic queue 'gamecenter.nfl.ravenssteelers.plays', i.e. a topic at the same

hierarchical level as 'gamecenter.nfl.ravenssteelers.statistics', the data payload for 'gamecenter.nfl.ravenssteelers.statistics' would not be placed in this fourth topic queue.").

102. Additionally, Spivey discloses "a plurality of aggregator components." *Id.* ¶108, Fig. 14. The aggregator components (1460-m) may subscribe to any of the topic queues as discussed above, for example, "gamecenter.nfl.ravenssteelers.plays." *Id.* Among other features, the aggregator component (1460) may consolidate data from multiple topic queues "into an aggregated topic … which is then published" and can be created "by an operator of the content provider entity for use by subscribing operators" or "subscribing operators may be permitted to suggest or create their own aggregated topics." *Id.* ¶108. Aggregator components also may provide data storage, operate as a backup publisher, and determine if the incoming real-time data has "updated." *Id.* ¶¶115, 116, 118, 122.

103. User devices connect to the system using an "address," which "may resemble a uniform resource locator (URL), for example 'torq.cbssports.com/torq.'" *Id.* ¶82, Fig. 8. Additionally, Figure 9 below shows that users can select "topics of live data to subscribe to" from a list of "topics available from a message queue server in a topic selection list 902:"

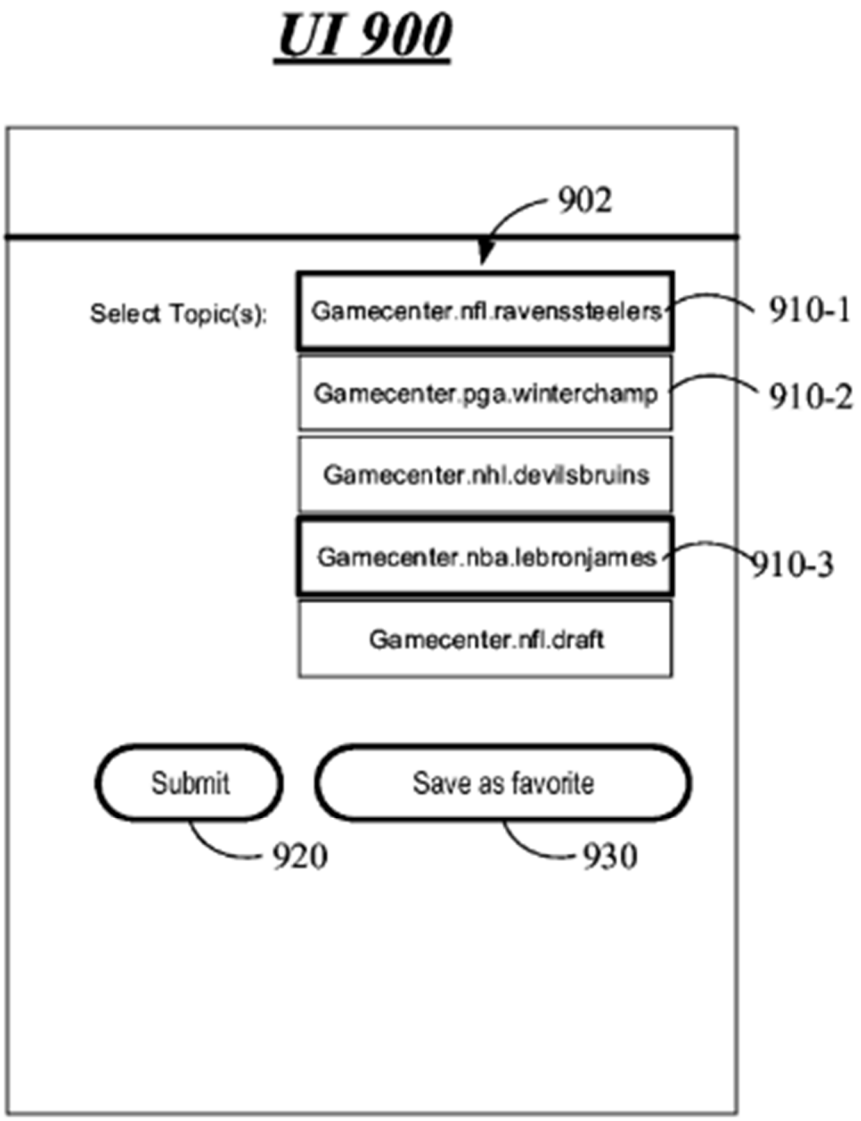


FIG. 9

Id., Fig. 9.

104. Spivey discloses that the connections between the system and the user are "full-duplex connection[s], such as a websocket connection" or that a "half-duplex connection may be utilized as well." *Id.* ¶61. A websocket connection "typically comprises a two-way connection between two entities, where each of the

two entities can send and receive data simultaneously." *Id*. A POSA would have understood that—in implementing such a "websocket connection"—the MQSD content distribution server would constitute a socket server. Finally, data is sent to connected and subscribed user devices from the topic queues. *Id.* ¶46 ("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.").

IX. A POSA'S MOTIVATION TO COMBINE PRIOR ART REFERENCES WITH A REASONABLE EXPECTATION OF SUCCESS

A. Ellis in View of Spivey

1. *Ellis*

105. In my opinion, a POSA would have been motivated to combine the teachings of Ellis and Spivey. As described above in §VIII.B, Ellis is generally directed to systems and methods that enhance "sports-related media content access and display." Ex.1016, Abstract. Ellis discloses an interactive system for watching sports content, including "wagering and fantasy league support." *Id.* More specifically, Ellis is directed to a two-part delivery method/system comprising transmitting a video stream originated by a content source (*i.e.*, "media source") over communication path (150) and transmitting a data stream originated by a SCS over a separate communication path (152), as shown below:

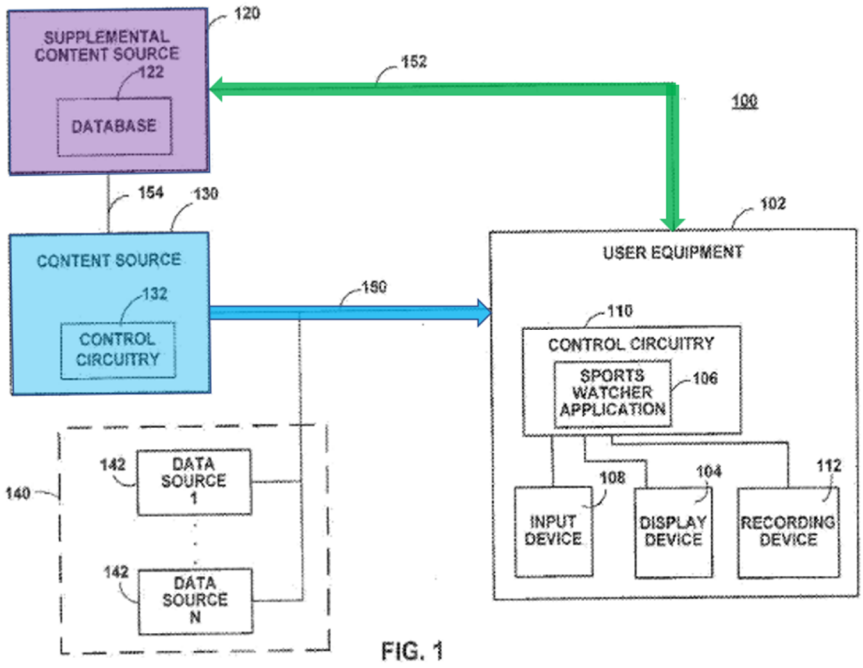


FIG. 1

Ex.1016, Fig. 1. Ellis teaches that the SCS "may be any suitable content source" that provides live data feeds containing "score, play, and up-to-date game summary information to user equipment." Ex.1016 ¶¶41, 155. Although Ellis discloses the claimed functionality of delivering live video and real-time supplemental content data over separate channels, Ellis does not explicitly disclose the backend architecture/process flows for delivering the supplemental content data.

2. Socket Server (Spivey)

106. As I explained in §VIII.C, Spivey provides details relating to backend server architecture supporting a live-event data delivery system. Spivey explains that some live-event data delivery systems, such as the system disclosed in Ellis, can suffer from latency issues. Spivey improves on those systems by teaching the use

of a control server and a socket server to provide live-event data to any number of customers in milliseconds. Ex.1017 ¶¶27-28.

107. In my opinion, a POSA—looking to improve Ellis's latency issues—would have been drawn to a system such as Spivey's and would have recognized the potential in combining these references to achieve full-duplex communications between a client and a remote host while minimizing latency. *See* Ex.1046 at 46, 49-52. After evaluating Ellis and Spivey as a whole from the perspective of a POSA, I conclude that it would have been well within the purview of a skilled artisan to integrate the websocket server into Ellis' digital content delivery system. A POSA would have been motivated to do so for the purpose of enabling the digital content delivery system described by Ellis with while minimizing latency between consumers.

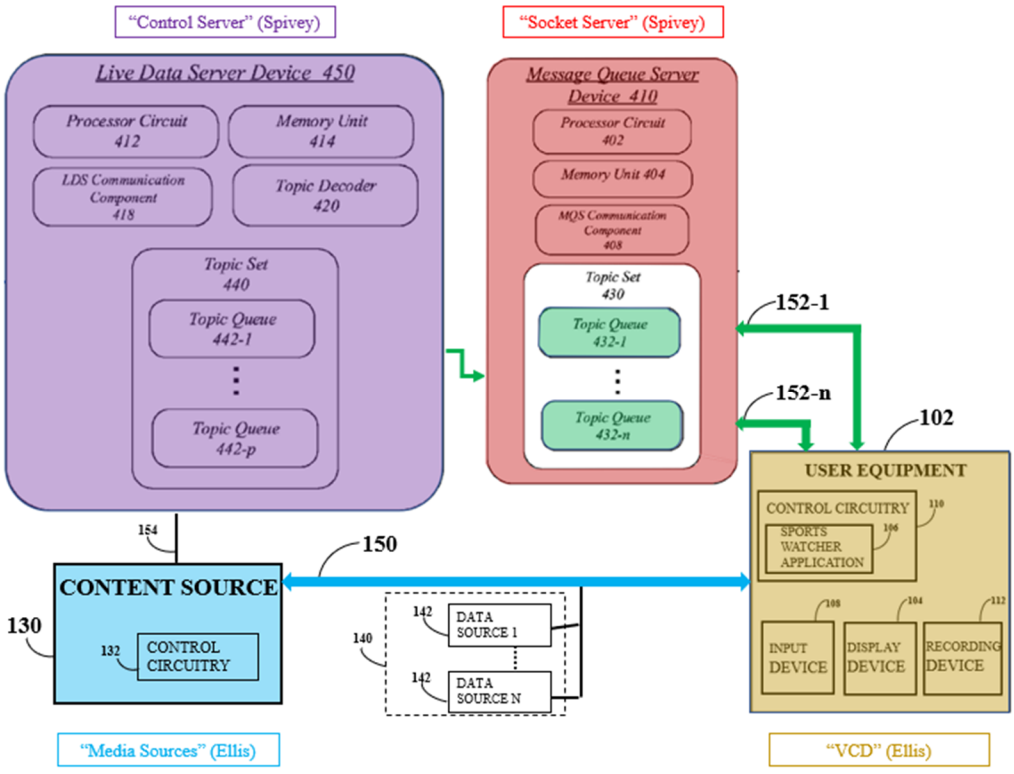
108. A POSA would have understood that socket servers were well-known components for implementing real-time data delivery, such as in Ellis, with reduced latency. §I.V.C.3. *supra*; Ex.1017 ¶¶96 (describing pushing a data payload from the message queue server device, *i.e.*, a socket server, to the client device "with a time delay of 100 milliseconds or less"), ¶35 (broadly describing a message queue server as "one or more electronic devices capable of receiving, processing and sending information to the system," including a wide range of exemplary devices). For example, a POSA would have appreciated Spivey's teaching that network

components—specifically the Live Data Server Device (LDSD) (*i.e.*, "control server") and Message Queue Server Device (MQSD) (*i.e.*, "socket server")—are useful in a data delivery system because they help dramatically reduce latency. Ex.1017 ¶¶28 (disclosing a system designed to reduce delay "to an order of milliseconds"), ¶43 (disclosing that a client device may request and participate in a websocket connection), ¶96 (reciting pushing digital content to a client device "substantially simultaneously … *e.g.*, with a time delay of 100 milliseconds or less … .").

109. Websockets protocol was an emerging technology that offered particular benefits for low-latency streaming when compared to the existing communication protocols. In my opinion, a POSA would have appreciated the benefits associated with the websocket connection that Spivey's socket server makes it possible to communicate with VCDs, including convenient and efficient full-duplex communications between a client and a remote host with low latency. *See* Ex.1017 ¶¶43, 61, 74, 92; Ex.1046 at 46. This is particularly true where Ellis's data delivery system does not specify a communications protocol. Thus, a POSA would have been motivated to integrate Spivey's LDSD and MQSD into Ellis's data delivery system to reduce latency to a time period "at least two orders of magnitude shorter than conventional time periods of legacy devices," as taught by Spivey and

to obtain additional benefits associated with websocket communications. Ex.1017 ¶96.

110. I believe that a POSA seeking to achieve the above-described results would have done so by implementing Spivey into Ellis's current data delivery system by replacing Ellis's SCS with Spivey's LDSD (*i.e.*, "control server") and adding Spivey's MQSD (*i.e.*, "socket server") to achieve a predictable result of decreased data delivery latency, as illustrated below in the annotated, combined diagram of Ellis Figure 1 and Spivey Figure 4 that I created for illustrative purposes:



See Ex.1016 ¶¶39, 41, Fig. 1 (recreated); Ex.1017 ¶¶53–57, Fig. 4 (recreated) (combined as explained above). The combined system would maintain the video

content delivery system and upgrade the supplemental content data delivery system by including a socket server to provide real-time data with reduced latency.

111. It should be noted that a socket server is a fundamental element of Internet design - it allows for the design of applications without the concern and technical detail of incorporating the lower layers of an Internet communication system. It cuts to the heart of an Internet communication system, as it allows for creativity at the application layer without having to incorporate the details of the transport, network, data link, and physical layers.

112. In my opinion, a POSA would have been motivated to modify Ellis in view of Spivey because both references relate to the same technical subject matter—content delivery for real-time events. *Cf.* Ex.1016, Abstract, *with* Ex.1017, Abstract. Both Ellis and Spivey describe methods and/or systems for streaming video and/or event data to one or more users, such that video and/or event data are transmitted simultaneously. *See, e.g.*, Ex.1016 ¶156; Ex.1017, Abstract. Further, I believe a POSA would have also combined Ellis and Spivey because the references solve the same or similar problems, namely, providing video and/or event data to users over the Internet quickly to multiple users. In my opinion, 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 systems for transmitting live video and related event data to viewers over the Internet. *See, e.g.*, Ex.1016 ¶¶40-41, 45, 74. Ellis also indicates that multiple users may view one or more different live events and receive related event information and discloses various systems for transmitting such information. *See, e.g.*, Ex.1016 ¶¶39-41, 45, 74.

113. Implementing certain portions of Spivey's system into Ellis's system would have been obvious because doing so would have required only the simple substitution of known elements for another to obtain predictable results. For example, Ellis discloses compatible hardware and software components to act as a source for event information to be transmitted to users. *See, e.g.*, Ex.1016 ¶47 ("Content source 130, supplemental content source 120, and data sources 140 are shown in FIG. 1 as separate elements" but "[i]n practice, their functionality may be combined into a single mechanism and provided from a single system at a single facility, or their functionality may be provided by multiple systems at multiple facilities."). 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.

114. In my opinion, a POSA would have been further motivated to combine Ellis with Spivey because Spivey provides a complementary solution describing the details of the backend architecture for transmitting event data to multiple users. Ex.1017, Abstract. Therefore, A POSA would have had a reasonable expectation of success in integrating Spivey's control server and socket server into Ellis's data delivery system. Understanding the compatibility of these similar systems, I believe a POSA would have had a reasonable expectation of success in integrating Spivey's system of "topic queues and open duplex communication connections" employing websockets to transmit "data from a live event (e.g., news event, a gaming event, sporting event, etc.)" to multiple users "in real-time" based on my knowledge and experience in the field. Ex.1017 ¶27.

B. Ellis in View of Spivey and Kellicker

115. The motivation to combine Ellis with Spivey is discussed above. *See* §IX.A.

116. Kellicker teaches a system to "[coordinate] data analytics to improve performance of streaming servers, client devices, and analytics servers." Ex.1049, ¶4. An exemplary system disclosed by Kellicker is illustrated below.

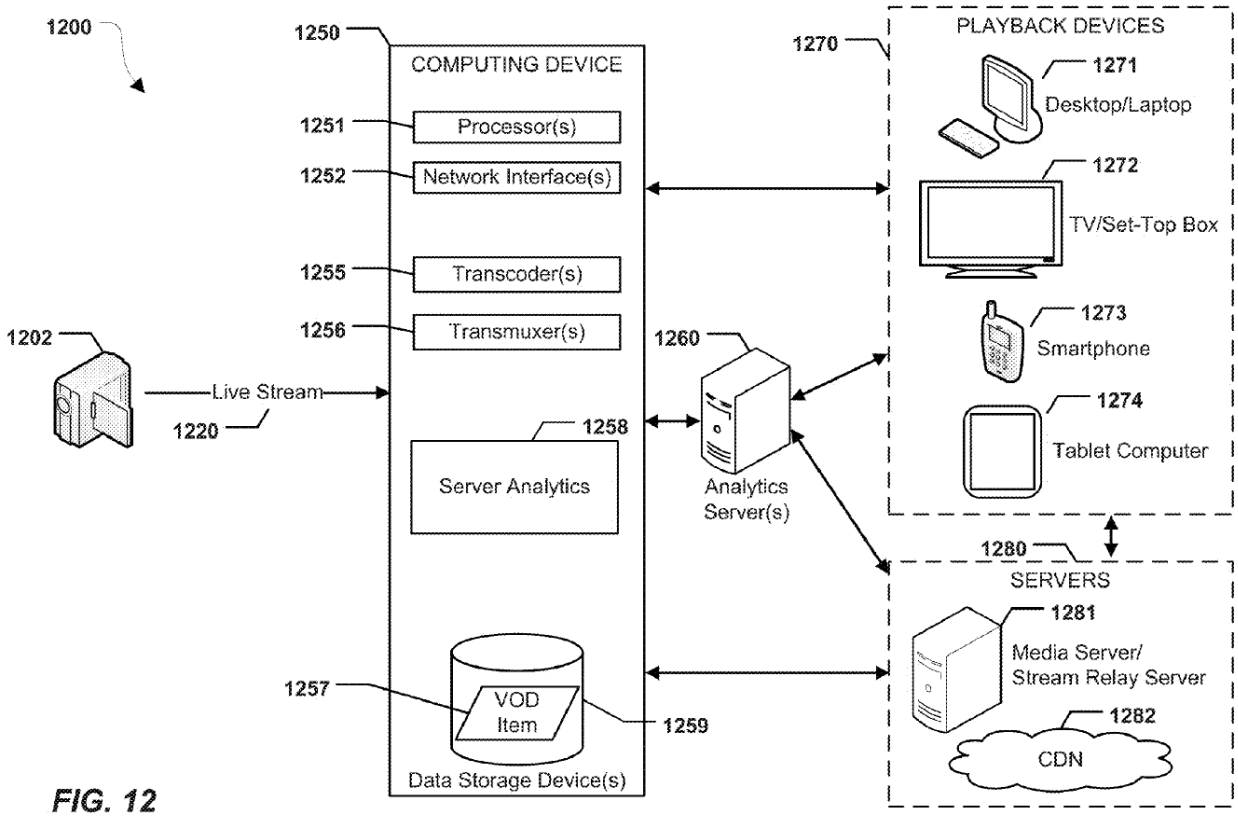


FIG. 12

Id., Fig. 12.

117. Based on at least the figure above and Kellicker's teachings, a POSA would have understood Kellicker's "computing device" 1250 to be analogous to both the "media sources" of the Challenged Patents', and the content sources 130 of Ellis.

118. Kellicker's computing device 1250 is a server. The server 1250 includes a processor, a memory and a network interface. Ex.1049, ¶¶20, 98-102; Fig. 12. Computing device 1250 receives incoming live media streams 1220 from a capture source (*e.g.*, broadcasting client) 1202, processes the incoming media and

serves it to playback devices 1270 either directly, or indirectly through other servers 1280.

119. Kellicker teaches providing server 1250 with the ability to support a number of real-time data communication protocols, including HLS, RTMP and/or WebRTC. *Id*., ¶106. Kellicker discloses that 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").

120. In my opinion, a POSA would have been motivated to further combine the teachings of Kellicker with Ellis and Spivey because all three references relate to the same technical subject matter—media content delivery for real-time events, such as live streaming. Ellis, Spivey and Kellicker all describe methods and/or systems for streaming video and/or event data to one or more users/client devices. In my opinion, a POSA would have understood that Ellis, Spivey and Kellicker solve the same or similar problems, namely, providing video and/or event data to multiple users/devices, and provide the ability to support the processing and transmission of media streams utilizing a number of different well-known protocols. In my opinion, a POSA would also have been motivated to further combine the teachings of Kellicker with Ellis and Spivey because Kellicker provides a complementary solution describing additional details regarding providing support for a number of

different real-time data communication protocols through one or more servers, including HLS, RTMP and/or WebRTC. In my opinion, a POSA would have successfully combined Ellis, Spivey and Kellicker, and the modifications proposed and discussed herein would have involved only the substitution or addition of known elements in a predictable manner to achieve a known or desired result. For example, in my opinion, a POSA would have found it obvious to integrate one or more media content servers implementing particular real-time data communication protocols, including an HLS server, RTMP server and/or WebRTC server, as taught by Kellicker, into the video content and event data delivery system taught by Ellis and Spivey. In my opinion a POSA would have had a reasonable expectation of success in doing so based on my knowledge and experience in the field.

C. Ellis in View of Spivey, Kellicker, and Phillips

121. The motivation to combine Ellis with Spivey and Kellicker is discussed above. *See* §IX.A-B.

122. Kellicker teaches an HLS caching server architecture (e.g., 1280/1282) communicatively coupled to at least one RTMP media server (e.g., 1250) and/or at least one WebRTC media server (e.g., 1250). Kellicker also teaches that this server architecture (e.g., 1280) may include a "content distribution network (CDN) 1282." Ex.1049, ¶102.

123. While Kellicker does not expressly describe the CDN architecture 1282, a conventional prior art CDN architecture typically includes both "parent" and "children" (e.g., edge or point of presence (PoP)) servers. As an example, a 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.

124. Phillips teaches that one of the objectives of its CDN architecture is to "[optimize] defragmentation of content in a content delivery network (CDN)." *Id*., Abstract. Consistent with this goal, Phillips provides 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." Ex.1050, ¶31. Phillips teaches that this architecture provides "for high-performance streaming . . . to subscribers using one or more Internet-based infrastructures." *Id*., ¶30. Fig. 12 below depicts an exemplary CDN architecture taught by Phillips.

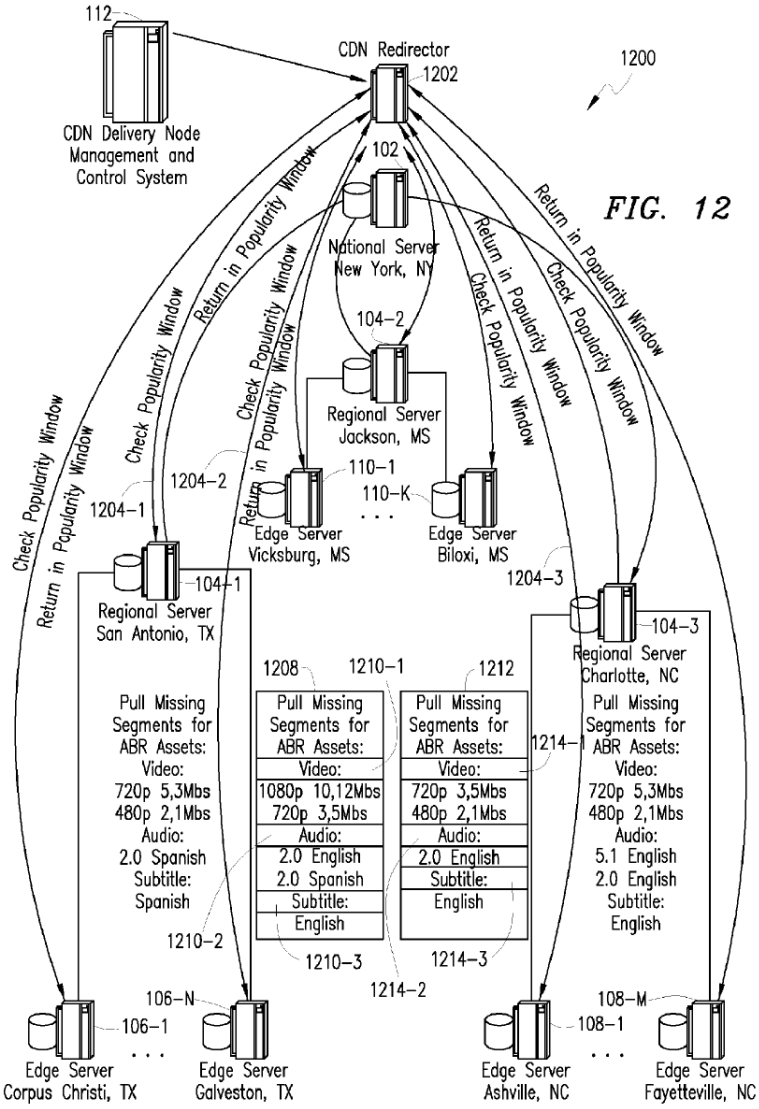


FIG. 12

125. 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, 31 (also disclosing that content may also come from other children nodes). In my opinion, a POSA would have been further motivated to incorporate Phillips' parent/child HLS caching server CDN architecture into the live streaming and event data delivery system taught by Ellis, Spivey, and Kellicker,

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 . . ."). A POSA would have understtod that pulling content from a parent node or other node in the CDN provides efficiency, for example, by distributing the request load throughout the CDN and limiting requests to a primary server, such as server 1250 in Kellicker. This increased efficiency represents a typical tradeoff with storage space/hardware (e.g., in the form of more servers). In my opinion a POSA would have been further motivated to incorporate Phillips' CDN architecture to achieve the benefits associated with optimizing the "defragmentation of content" in a CDN expressly taught by Phillips.

126. In my opinion, a POSA would have been motivated to further combine the teachings of Phillips with Ellis, Spivey, and Kellicker because all of these references relate to media content delivery systems including real-time streaming of live or real-time media. Ellis, Spivey, Kellicker, and Phillips all describe methods and/or systems for streaming video and/or event data to one or more users. In my opinion, Ellis, Spivey, Kellicker, and Phillips are directed to providing solutions to 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 media streams utilizing a number of different well-known protocols. In my opinion, a

POSA would also have been motivated to further combine the teachings of Phillips with Ellis, Spivey, and Kellicker because Phillips provides a complementary solution and describes additional details regarding an HLS caching server CDN architecture, such as is taught by Ellis, Spivey, and Kellicker. In my opinion, a POSA would have successfully combined Ellis, Spivey, Kellicker, and Phillips and the modifications proposed and discussed herein would have involved only the substitution or addition of known elements in a predictable manner to achieve a known or desired result. For example, in my opinion, a POSA would have found it obvious to simply substitute some or all of Phillips' parent/child HLS caching server CDN architecture with the CDN taught by Ellis in view of Spivey and Kellicker. In my opinion a POSA would have had a reasonable expectation of success in doing so based on my knowledge and experience in the field.

D. Ellis in View of Spivey, Kellicker, Phillips, and Stoica

127. The motivation to combine Ellis with Spivey Kellicker, and Phillips is discussed above. *See* §IX.A-C.

128. Stoica is directed to methods and arrangements for using networks, such of the Internet, and in a particular CDNs, to access content including video files and live streaming via client devices. Ex.1051, 1:15–17. Stoica discloses a video serving method and arrangement making use of the HTTP chunking delivery

protocol (e.g., HLS). *Id.*, 5:55–60. An example of such an arrangement is depicted in FIG. 8 below:

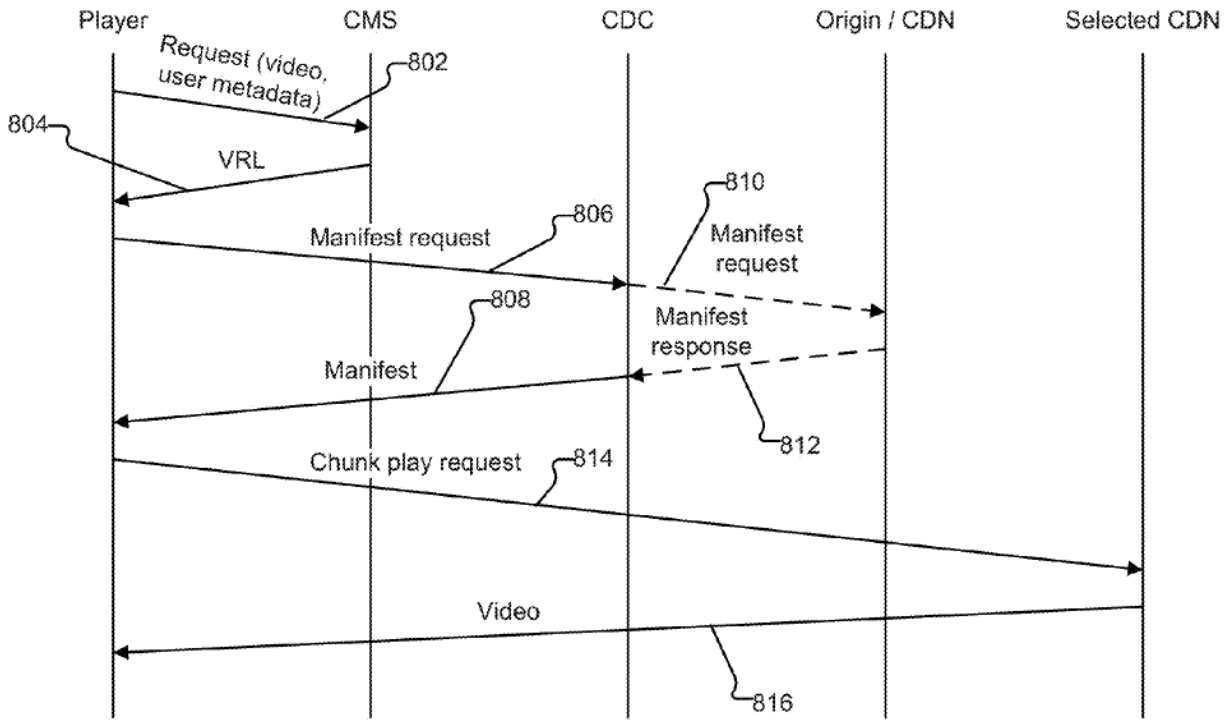


FIG. 8

129. Stoica's Content Management System (CMS) and Content Distribution Coordinator (CDC) are web servers that facilitate streaming of various media including live streams as an intermediary between the client (player) and the 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.

130. 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 requests, data accesses, etc. *Id.*, 9:63–10:5;11: 17–32. The arrangement is illustrated Fig. 11, reproduced below.

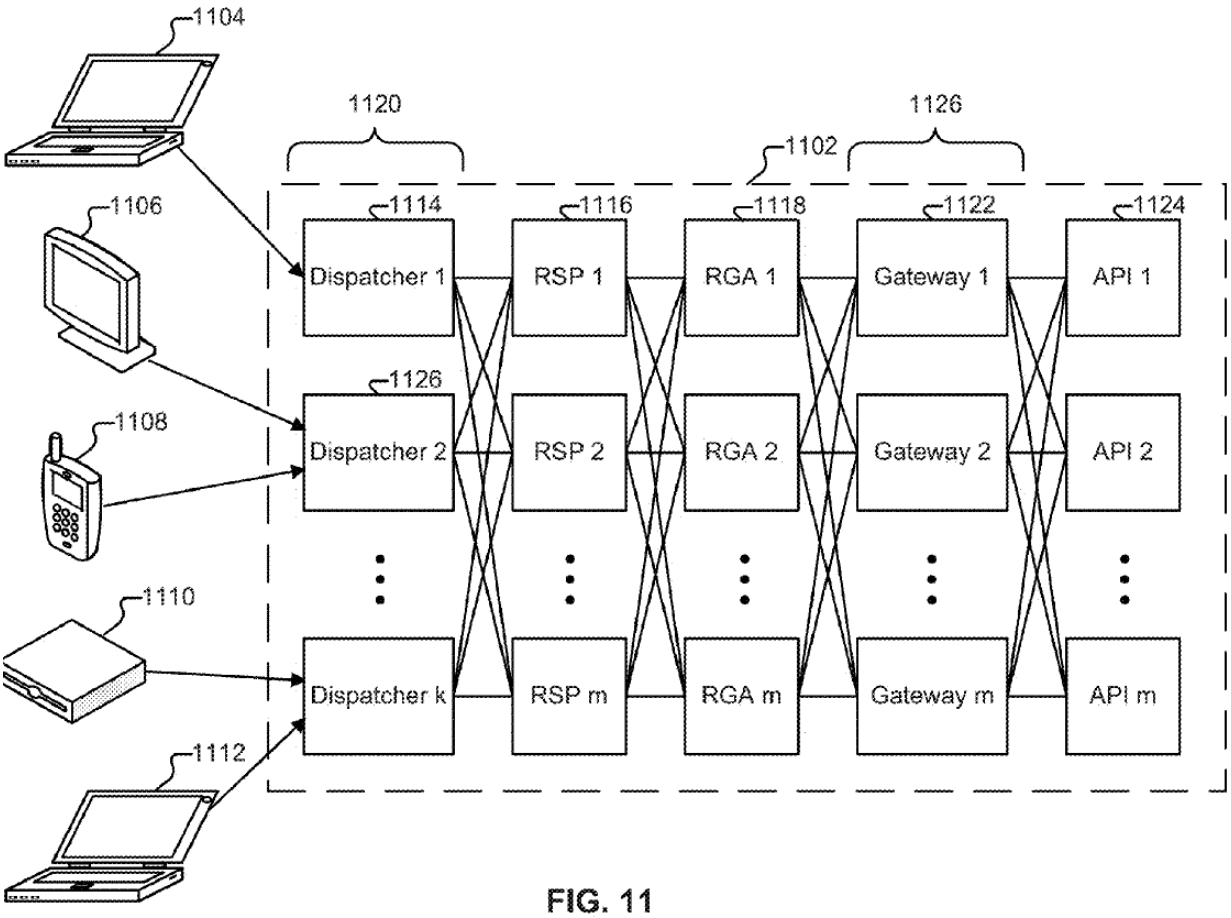


FIG. 11

131. In my opinion, it would have been obvious to a POSA to have implemented Stoica's web server architecture in order to facilitate communications and delivery of streaming media content between clients and the content source. In

my opinion, a POSA would have been motivated to do so at least because Stoica teaches addressing demand for efficient distribution of online media content, including video, with high quality. Ex.1051, 1:18–20; 9: 63–10:2; Fig. 11.

132. In my opinion, a POSA would have been motivated to further combine the teachings of Phillips with Ellis, Spivey, and Kellicker because all of these references relate to media content delivery systems including real-time streaming of live or other media (e.g., VOD). Ellis, Spivey, Kellicker, Phillips, and Stoica all describe methods and/or systems for streaming video and/or event data to one or more users. In my opinion, a POSA would also have been motivated to further combine the teachings of Stoica with Ellis, Spivey, Kellicker, and Phillips because Stoica provides a complementary solution and describes additional details regarding the implementation of communications among various nodes and components of, and the delivery of content and data through, a CDN architecture, such as is taught by Ellis in view of Spivey, Kellicker, and Phillips. In my opinion, a POSA would have understood that these 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 media streams utilizing a number of different well-known protocols. In my opinion, a POSA would have successfully combined Ellis, Spivey Kellicker, Phillips, and Stoica and the modifications proposed and discussed herein would have involved only the substitution or addition of known

elements in a predictable manner to achieve a known or desired result. For example, in my opinion, a POSA would have found it obvious to simply incorporate (or substitute) some or all of Stoica's servers (*e.g.*, the CMS, CDC, dispatcher server, etc.) and related functionality (*e.g.*, APIs, URLs, content identifiers, etc.) with the CDN architecture taught by Ellis in view of Spivey, Kellicker, and Phillips. In my opinion a POSA would have had a reasonable expectation of success in doing so based on my knowledge and experience in the field.

E. Ellis in View of Spivey, Kellicker, Phillips, and Pantos

133. The motivation to combine Ellis with Spivey Kellicker, and Phillips is discussed above. *See* §IX.A-C.

134. 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 the use of 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.)". ¶

135. Pantos is a version of the HLS streaming protocol dated April 4, 2016, and originally developed by Apple in 2009. Pantos describes the HLS 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. Because Pantos is a specification for HLS, a POSA would have looked to Pantos when implementing HLS as taught by Kellicker and Phillips.

136. Pantos teaches that the HLS protocol:

> provides 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 . . .
> *Id*., § 1.

137. In my opinion, a POSA would have been motivated to further combine the teachings of Pantos with Ellis, Spivey, Kellicker, and Phillips because all of these references relate to media content delivery including real-time streaming of live or other media (e.g., VOD). Ellis, Spivey, Kellicker, Phillips, and Pantos all describe methods and/or systems for streaming video and/or event data to one or more users. And, in particular, Pantos, like Kellicker and Phillips specifically relates to media content delivery through an HLS communications protocol. In my opinion, a POSA would also have been motivated to look to Pantos and further combine its teachings with Ellis, Spivey, Kellicker, and Phillips because Pantos provides a complementary solution and describes specific details regarding the implementation of an HLS CDN architecture, such as is taught by Ellis in view of Spivey, Kellicker, and Phillips. In my opinion, a POSA would have understood that these 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 media streams utilizing a number of different well-known protocols. In my opinion, a POSA would have successfully combined Ellis, Spivey Kellicker, Phillips, and Pantos, and the modifications proposed and discussed herein would have involved only the substitution or addition of known elements in a predictable manner to achieve a known or desired result. For example, in my opinion, a POSA would have found it obvious to simply incorporate (or substitute) some or all of the elements and features of the HLS specification taught by Pantos (*e.g.*, specific contents, features, and attributes of the HLS playlists, file segments and video segments, etc.) with the HLS CDN caching server architecture taught by Ellis in view of Spivey, Kellicker, and Phillips. In my opinion a POSA would have had a reasonable expectation of success in doing so based on my knowledge and experience in the field.

F. Herzog in View of Spivey and Ellis

138. Herzog discloses methods and systems for viewing multiple videos related to the same sporting event in addition to live information about the event (meta-data). *See* VIII.A. For example, Herzog's Figure 7 shows "four different videos of a live event and meta-data of multiple parameters" from which a user can select:

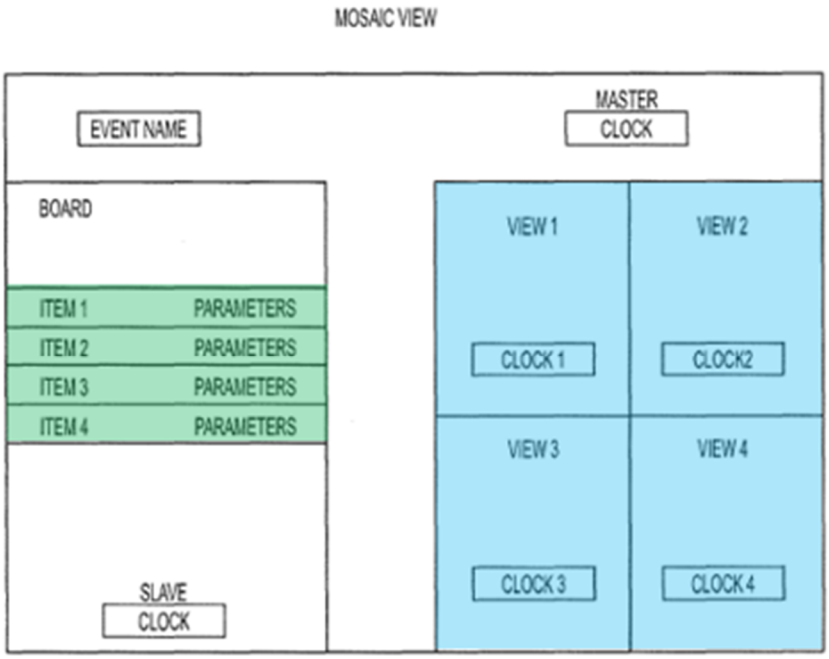


FIG. 7

Ex.1015 ¶¶12, 80-81, Fig.7 (annotated).

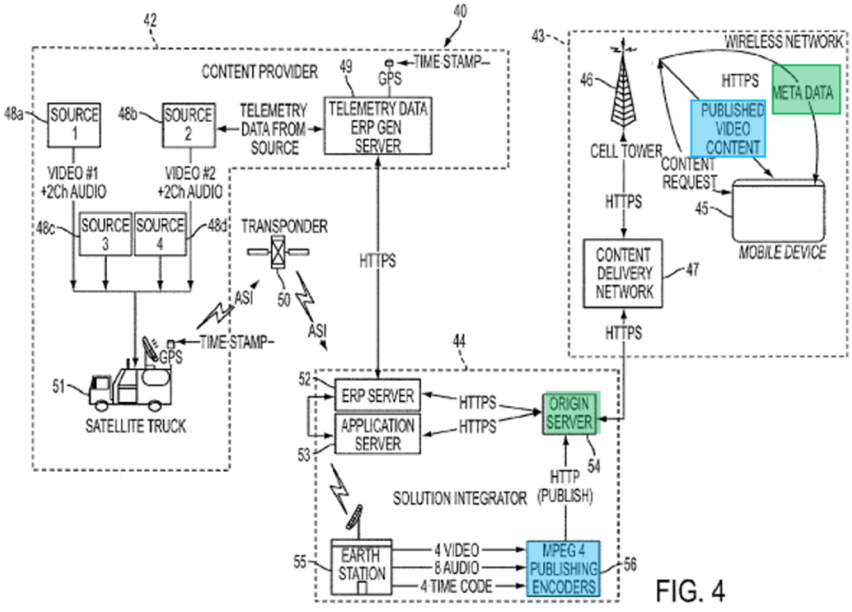


FIG. 4

Ex.1015 ¶¶48-52, 67, Figs. 4 (annotated), 6. The video and data streams can be provided over separate connections. Ex.1015, Fig. 6; ¶¶70–77, 83.

1. Socket server (*Spivey*)

139. Herzog does not specifically disclose a backend server architecture supporting the data distribution—the claimed socket server. However, it is my opinion that a POSA would have known that the websocket client-server communication protocol employed by Spivey was an alternative to the HTTP polling used by Herzog. *See* Ex.1046 at 46-47. Therefore, a POSA would have found it obvious to implement Herzog's Origin Server as a websocket server—as taught by Spivey—to achieve full-duplex communications between a client and a remote host with low latency. *Id*. at 46 (observing that a websocket protocol provided a three-to-one reduction in latency as compared to half-duplex HTTP polling).

140. As noted above, it is my opinion that a POSA would have understood that socket servers were well-known components for implementing real-time data delivery, such as in Herzog, with reduced latency. *See* §IV.C.3. *supra*; Ex.1017 ¶¶35, 96. In my opinion, a POSA also would have appreciated the benefits associated with Spivey's MQSD (*i.e.,* "socket server") which establishes websocket connections with client devices, including convenient and efficient full-duplex communications between a client and a remote host with low latency. Ex.1017 ¶¶27-28, 43, 61, 74, 92, 96; Ex.1046 at 46; §IX.A. Therefore, a POSA, looking to improve Herzog's latency issues, would have looked to implement a system such as Spivey's.

141. In my opinion, a POSA would have looked to improve Herzog's latency issue by incorporating Spivey into Herzog's current data delivery system by implementing Spivey's MQSD at Herzog's Origin Server to achieve a predictable result of decreased latency data delivery. *See* Ex.1015 ¶¶50-51, Fig. 4; Ex.1017, Fig. 4. The combined system would maintain Herzog's video content delivery system and upgrade the data delivery system to reduce latency.

142. In my opinion, a POSA would have been motivated to combine Herzog with Spivey for many of the same reasons that a POSA would have been motivated to combine Ellis with Spivey. First, Herzog and Spivey relate to the same technical subject matter—content delivery for real-time events. *See* Ex.1017, Abstract; Ex.1015, Abstract. Both references describe methods and/or systems for streaming video and/or event data to one or more users, such that video and/or event data are transmitted simultaneously. *See, e.g.*, Ex.1017, Abstract; Ex.1015, Abstract. Second, the references solve the same or similar problems, namely, providing video and/or event data to multiple users over the Internet quickly. *See, e.g.*, Ex.1017 ¶27; Ex.1015 ¶¶3, 19, 51.

143. As such, a POSA would have had a reasonable expectation of success in combining Herzog with Spivey because Spivey contemplates being implemented in a digital content delivery system such as Herzog, stating that it can be used with "any well-known communication techniques and protocols, such as techniques

suitable for use with packet-switched networks (e.g., public networks such as the Internet, private networks such as the enterprise intranet, and so forth) ... ." Ex.1017 ¶155. In my opinion, a POSA would have been motivated to combine Herzog with Spivey because Spivey provides a complementary solution describing the details of the backend architecture for organizing and transmitting event data to multiple users. Ex.1017, Abstract.

Combining certain portions of Herzog and Spivey would have been obvious because doing so would require only the simple substitution of known elements for another to obtain predictable results. Herzog encourages this flexibility, stating that it encourages "modifications … in various forms and examples, and that the teachings may be applied in numerous applications." Ex.1015 ¶90. To a POSA, it would have been obvious to substitute any of Herzog's meta-data distribution 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.

2. Providing Users with Chat, Sports Score, and Sports Commentary Information Using Separate Communication Channels (Ellis)

144. As explained above, Herzog in combination with Spivey discloses a system providing video streams and event information streams to multiple users. *See* §IX.C. To the extent Herzog in combination with Spivey does not disclose a third

stream—the claimed real-time information stream—a POSA would have found it obvious to implement such a stream into Herzog/Spivey, in view of Ellis.

As set forth above, Ellis discloses a similar content delivery system configured to deliver live video and data over separate channels. *See* §VIII.B and IX.A. Ellis discusses transmitting the video "broadcasts" over a communication path and "game summary information, … and any other sports-related information" over another communication path. 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." *Id*. Thus, it is my opinion that a POSA would have understood that Ellis teaches transmitting via a first real-time information Internet communication channel between a real-time information socket of the at least one socket server and the first VCD.

a. Chat

145. Ellis further teaches that its system is configured to transmit chat information. Ex.1016 ¶132 (users "may also leave feedback regarding the commentary that other users in the media system may view when they access the same commentary."). In my opinion, a POSA would have understood that Ellis discloses a system that transmits and receives real-time information, such as chat information, regarding the live stream of digital content. *See* Ex.1016 ¶126 ("To

create and send a message (e.g., instant message or email message), the user may select send message option.").

146. Herzog in view of Spivey and Ellis discloses a socket specifically designated for chat information. Spivey utilizes topic queues to organize "data payloads" that "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." 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. 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." In sum, the statistics would be separate from the chat information. Thus, it is my opinion that a POSA would have understood Spivey to disclose a real-time information socket corresponding to the real-time information, such as chat information, relating to the digital content.

147. As such, a POSA would have been motivated to integrate Ellis's additional real-time information apparatus into Herzog/Spivey's system. In my opinion, a POSA would have been motivated to combine Ellis with Herzog/Spivey because Ellis builds on Herzog's functionality by explicitly disclosing providing

more content options, *e.g.*, real-time information, such as chat information or advertising content, to a content distribution system such as Herzog: "[C]ontent source 130 may be any suitable source of media content, such as … [an] Internet or network media/webserver, or any other … system for originating and distributing … content to user equipment 102" that provides "broadcast programming" and "sporting events." Ex.1016 ¶¶39-41, 62, Fig. 1, 2B. Moreover, a POSA would have understood that users would enjoy the interactive aspect that Ellis's real-time information stream would offer.

148. For a POSA, adding an additional real-time information stream to a system such as Herzog (among other content types), which already provides multiple live streams to clients, would have involved nothing more than applying a known technique to a known device that was ready for improvement to yield a predictable result. A POSA would have had a reasonable expectation of success in doing so because Ellis specifically contemplates being implemented in a digital content delivery system such as Herzog, stating that it can be used with "any suitable source of media content," which includes IPTV content "delivered on a particular network address (e.g., multicast address) on which an IP socket is currently joined." Ex.1016 ¶¶39-40, 74.

b. Sports Score and Sports Commentary Information

149. Ellis is concerned with systems and methods "for providing enhanced information access and viewing functionality for sports -related programming." Ex.1016, at ¶8. As one way of achieving this objective Ellis teaches that "sports score . . . information may be included in one real-time data feed, while sports commentary may be included and another feed." *Id*., ¶41.

150. As a POSA would have recognized, Herzog was already equipped to provide multiple, different live video and meta-data streams, so configuring this system to provide sports score and sports commentary information/data in separate data feeds would have been obvious and well within the skill of a POSA.

151. A POSA would have been motivated to configure Herzog with Ellis's functionality because Ellis teaches that by doing so information access and viewing functionality for the user is enhanced. A POSA would have understood that users would enjoy having more content options, *e.g.*, sports commentary, score information, chat information, etc. Ex.1016 Abstract, ¶¶46, 68, Figs. 1, 2B, 3.

152. A POSA would have had a reasonable expectation of success in making such modifications, *i.e.*, providing separate feeds of sports score and sports commentary information, because Ellis specifically contemplates being implemented in a content delivery system such as Herzog, stating that it can be used with "any suitable source of media content," which includes IPTV content

"delivered on a particular network address (e.g., multicast address) on which an IP socket is currently joined." Ex.1016 ¶¶39-40, 74.

153. Furthermore, in my opinion, a POSA would have been motivated to combine Herzog with Spivey and Ellis because they relate to the same technical subject matter—content delivery for real-time events. *Cf.* Ex.1016, Abstract, *with* Ex.1017, Abstract, *and* Ex.1015, Abstract. All three references describe methods and/or systems for simultaneously streaming video and/or event data to one or more users. *See, e.g.*, Ex.1016 ¶156; Ex.1017 Abstract; Ex.1015, Abstract. 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.1015 ¶¶3, 19, 51. As such, it would have been obvious to a POSA at the time of the alleged invention to look to expand upon Herzog with Spivey and Ellis to arrive at the claimed invention.

G. Herzog in View of Spivey, Ellis, Kellicker, and Phillips

154. The motivation to combine Herzog with Ellis and Spivey is discussed above. *See* §IX.F.

155. Kellicker teaches a system to "[coordinate] data analytics to improve performance of streaming servers, client devices, and analytics servers." Ex.1049, ¶4. An exemplary system disclosed by Kellicker is illustrated below.

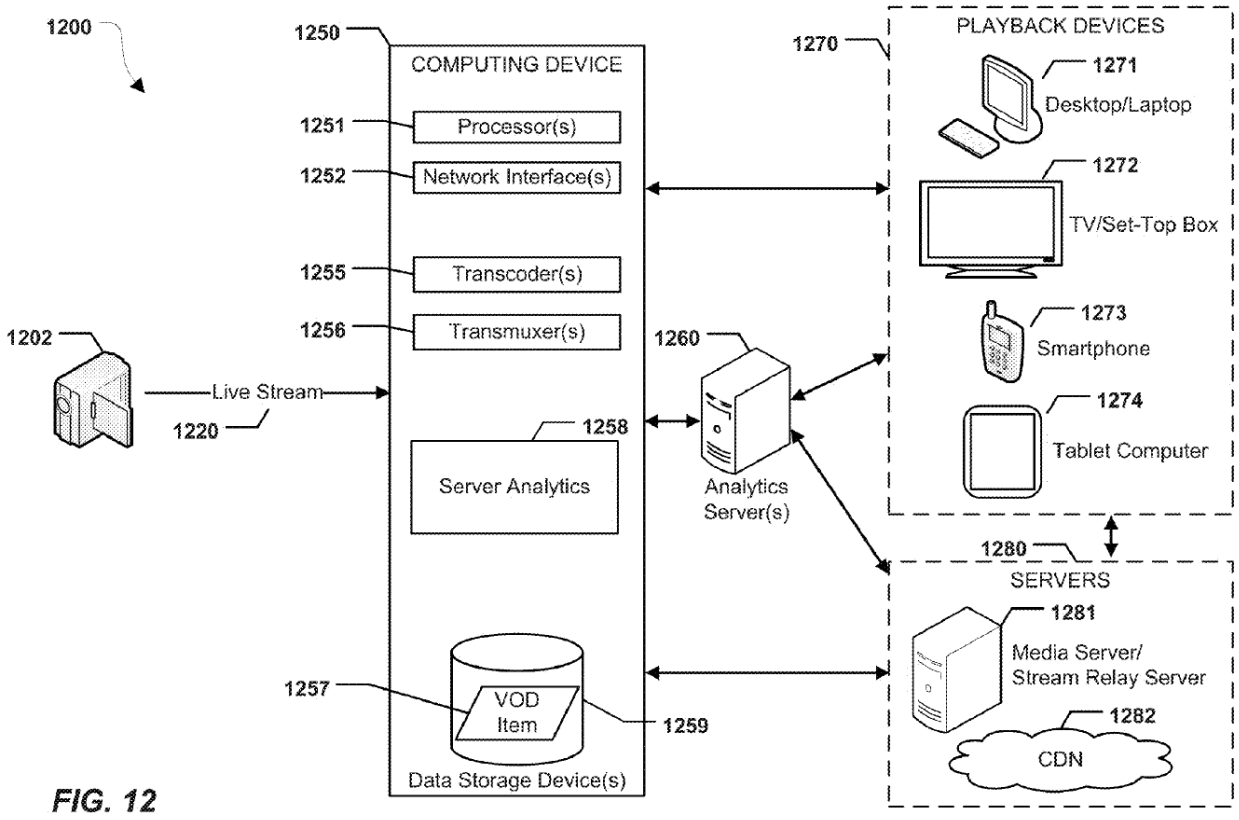


FIG. 12

Id., Fig. 12.

156. Based on at least the figure above and Kellicker's teachings, a POSA would have understood Kellicker's "computing device" 1250 to be analogous to both the "media sources" of 'the Challenged Patents, the content sources 130 of Ellis, and the audio/video streaming portion of Herzog's origin server. Herzog, ¶51 ("origin server 54 then transmits both the video/audio stream and the meta-data to the content delivery network 47"), ¶55 ("encoded video is then published to the content delivery network 47 (for example) via IP utilizing the HLS, smooth stream, or DASH encoding techniques with a time stamp wrapper"), FIG. 4.

157. Kellicker's computing device 1250 is a server. The server 1250 includes a processor, a memory and a network interface. Ex.1049, ¶¶20, 98-102; Fig. 12. Computing device 1250 receives incoming live media streams 1220 from a capture source (*e.g.*, broadcasting client) 1202, processes the incoming media and serves it to playback devices 1270 either directly, or indirectly through other servers 1280.

158. Kellicker teaches providing server 1250 with the ability to support a number of real-time data communication protocols, including HLS, RTMP and/or WebRTC. *Id*., ¶106. Kellicker discloses that 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").

159. While neither Kellicker nor Herzog expressly describe the CDN architecture, a POSA would have readily understood that a conventional prior art CDN architecture typically includes both "parent" and "children" (e.g., edge or point of presence (PoP)) servers. As an example, a 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.

160. Phillips teaches that one of the objectives of its CDN architecture is to "[optimize] defragmentation of content in a content delivery network (CDN)." *Id*., Abstract. Consistent with this goal, Phillips provides 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." Ex.1050, ¶31. Phillips teaches that this architecture provides "for high-performance streaming . . . to subscribers using one or more Internet-based infrastructures." *Id*., ¶30. Fig. 12 below depicts an exemplary CDN architecture taught by Phillips.

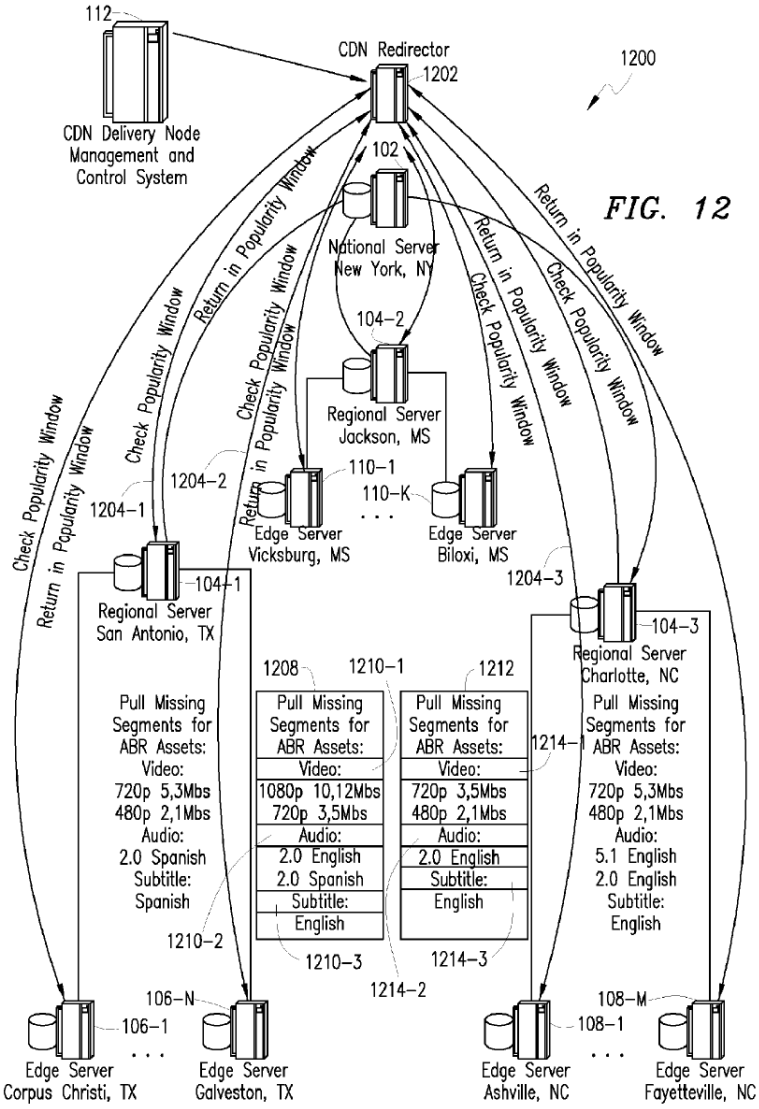


FIG. 12

161. 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, 31 (also disclosing that content may also come from other children nodes). In my opinion, a POSA would have been further motivated to implement Phillips' 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 . . .”). Pulling content from a parent node or other node in the CDN provides efficiency, for example, by distributing the request load throughout the CDN and limiting requests to a primary server, such as server 1250 in Kellicker. This increased efficiency represents a typical tradeoff with storage space/hardware (e.g., in the form of more servers).

162. In my opinion, a POSA would have been motivated to further combine the teachings of Kellicker and Phillips with Herzog, Ellis, and Spivey because all of these references relate to media content delivery systems including real-time streaming of live or real-time media. Herzog, Ellis, Spivey, Kellicker, and Phillips all describe methods and/or systems for streaming video and/or event data to one or more users. In my opinion, a POSA would also have been motivated to further combine the teachings of Kellicker and Phillips with Herzog, Ellis, and Spivey because Kellicker and Phillips provide complementary solutions and describe additional details regarding providing support for a number of different real-time data communication protocols through one or more servers, including HLS, RTMP and/or WebRTC, and the implementation of a parent/child HLS caching server architecture for a CDN, such as is taught by Herzog in view of Ellis and Spivey. In my opinion, Herzog, Ellis, Spivey, Kellicker, and Phillips are directed to providing solutions to 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 media streams utilizing a number of different well-known transmission protocols. In my opinion, a POSA would have successfully combined Herzog, Ellis, Spivey, Kellicker, and Phillips and the modifications proposed and discussed herein would have involved only the substitution or addition of known elements in a predictable manner to achieve a known or desired result. For example, in my opinion, a POSA would have found it obvious to integrate or simply substitute some or all of Kellicker's media content servers implementing particular real-time data communication protocols, including an HLS server, RTMP server and/or WebRTC server, and Phillips' parent/child HLS caching server CDN architecture with the video content and event data delivery system and CDN taught by Herzog in view of Ellis and Spivey. In my opinion a POSA would have had a reasonable expectation of success in doing so based on my knowledge and experience in the field..

X. GROUNDS

163. It is my opinion that the Challenged Claims of the Challenged Patents are invalid. I understand that in IPR proceedings, a petitioner may challenge the claims of a patent only on the basis of lack of novelty or obviousness based on patents and printed publications. Nothing in this Declaration should be construed as expressing any position or opinion as to whether the claims constitute patentable subject matter under 35 U.S.C. §101, or whether they satisfy the definiteness,

enablement, or written description requirements of 35 U.S.C. §112. I reserve the right to opine on other matters of invalidity in other proceedings, including in addition to anticipation and obviousness, lack of written description, lack of enablement, indefiniteness, non-patentable subject matter, and improper inventorship.

164. For convenience, I have summarized the various grounds of invalidity discussed in this Declaration below:

’697 and ’687 Patents			
Ground	**Basis**	**Claim(s)**	**Reference(s)**
1	§103	1	Ellis in view of Spivey
2	§103	10	Ellis in view of Spivey and Kellicker
3	§103	11 and 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
6	§103	23 and 24	Ellis in view of Spivey
7	§103	26	Ellis in view of Spivey, Kellicker and Phillips
8	§103	23 and 24	Herzog in view of Spivey and Ellis
9	§103	26	Herzog in view of Spivey, Ellis, Kellicker and Phillips

A. The '697

1. '697 Ground 1: Claim 1 is Obvious Over Ellis in View of Spivey

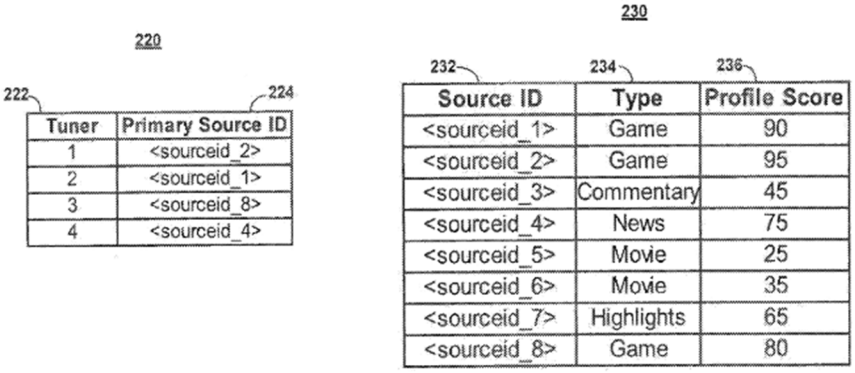
165. For the reasons stated in §IX.A. above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis and Spivey with a reasonable expectation of success to arrive at the claimed invention.

a. Independent Claim 1

i. **Element 1.0: "A system for providing a first plurality of copies of a first broadcaster's live stream of digital content including first live sporting event video-based commentary from a first broadcaster client device to a first plurality of viewer client devices, and for providing a second plurality of copies of a second broadcaster's live stream of digital content including second live sporting event video-based commentary from a second broadcaster client device to a second plurality of viewer client devices, the system comprising:"**

166. To the extent the preamble is limiting, it is my opinion that Ellis discloses this element. *See* Ex.1016, Abstract. Ellis discloses a system for providing a first/second plurality of copies of a first/second broadcaster's live stream of digital content including first/second live sporting event video-based commentary from a first/second broadcaster client device 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. Figure 2B, below, shows live sporting events available at the same time to any number of users.



220

222 Tuner	224 Primary Source ID
1	<sourceid_2>
2	<sourceid_1>
3	<sourceid_8>
4	<sourceid_4>

230

232 Source ID	234 Type	236 Profile Score
<sourceid_1>	Game	90
<sourceid_2>	Game	95
<sourceid_3>	Commentary	45
<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 those sporting events "to user equipment 102." *Id.* ¶38-40. Ellis itself defines "sporting event" as including "online sports commentary." *Id.* ¶38. And, media content includes broadcast programming "and any other content capable of being displayed by, presented to, … or interacted with, using user equipment 102." *Id.* ¶40. Also, Ellis teaches providing sports commentary in a data feed or over communications channel to a user. *Id.*, ¶41. It is my opinion that broadcasts of live events, including live sporting events, would obviously, if not inherently, include video-based commentary. Additionally, "commentary" is natively supported in HLS live streams as alternative audio to an associated video stream (i.e., video-based commentary). Ex.1052, 28-29 (§8.6).

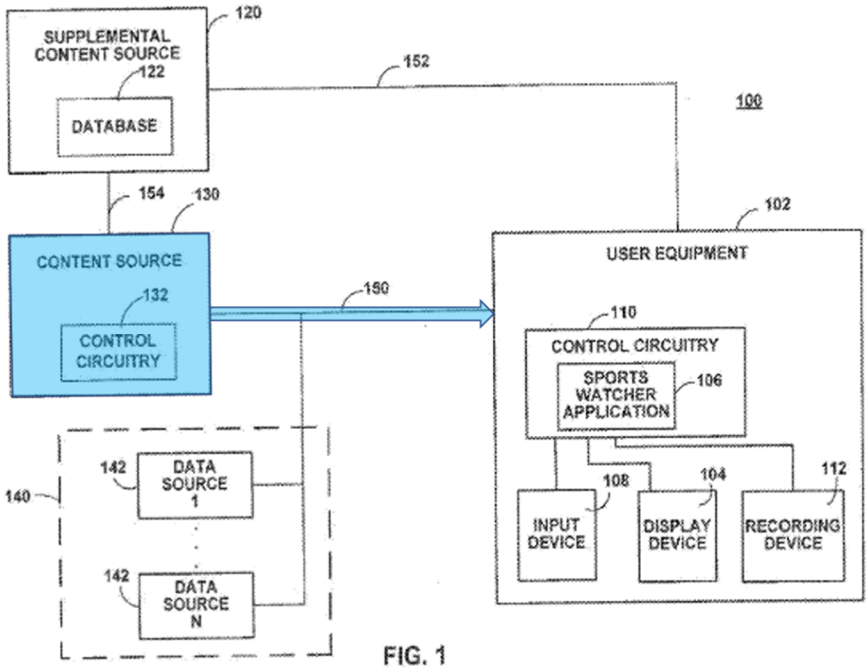
167. Ellis further discloses multiple broadcasters providing copies of each broadcaster's live stream of digital content including live sporting event video-based

commentary 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. For example, Ellis's system broadcasts digital streams from the content source, which can be an "[i]nternet or network media/webserver." 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. *Id.* ¶45, 74. The communication path can be any "suitable wired or wireless path." *Id.* "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. In my opinion, a POSA would have understood Ellis' distribution system provides for a plurality of broadcaster devices sending, a plurality of copies of multiple broadcaster's live stream of digital content to a plurality of viewers.

ii. **Element 1.1: "A) a plurality of media sources to:"**

168. In my opinion, Ellis discloses content sources, which are "any suitable source of media content" such as an "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 serve to 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. Figure 1 shows a content source distributing video content to users over communication path 150:



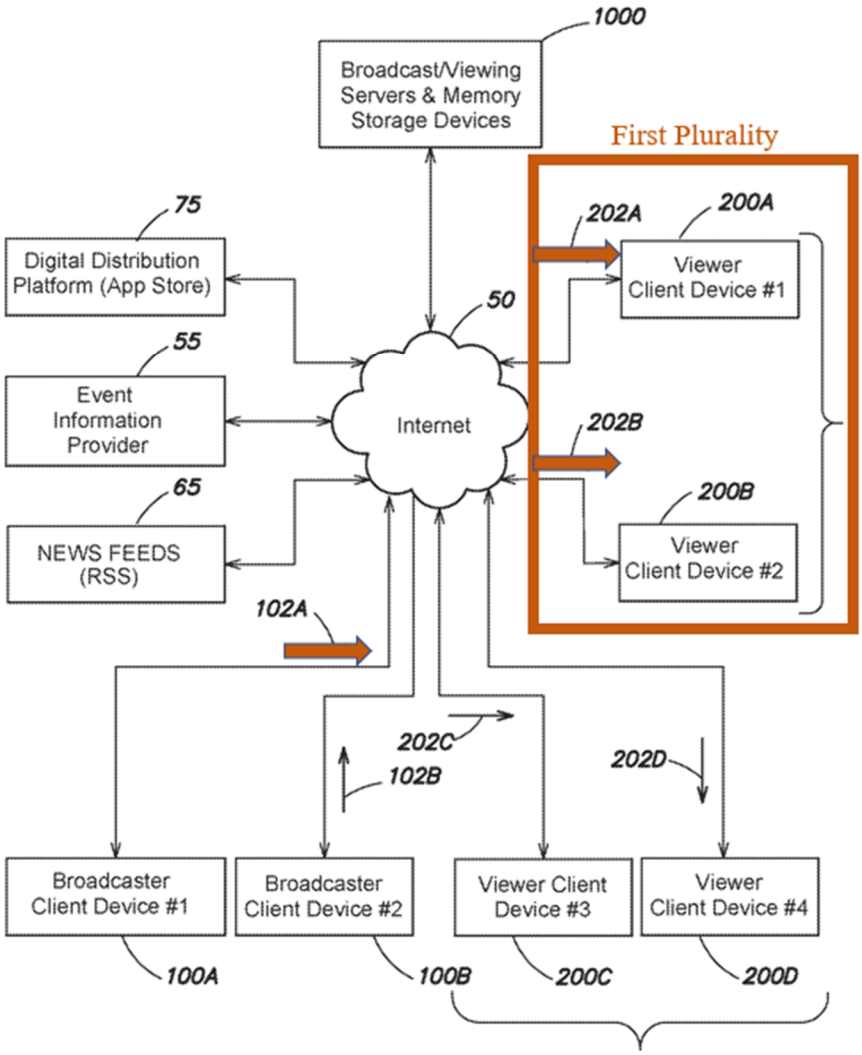
Ex.1016 ¶¶39-41, Fig. 1 (annotated). In my opinion, Ellis also discloses multiple content sources, *i.e.*, a plurality. *Id.* ¶39 ("[T]here may be numerous instances of content source 130 …"), 74. As shown in Fig. 1 above, Ellis also discloses that users can be connected with a plurality of data sources 140. *Id.* Moreover, as explained below, it is my opinion that Ellis's content sources perform the same functions as the media sources as recited in elements 1.2-1.4. Thus, a POSA would have understood that Ellis discloses the recited element.

iii. Element 1.2: "receive the first broadcaster's live stream of digital content and the second broadcaster's live stream of digital content;"

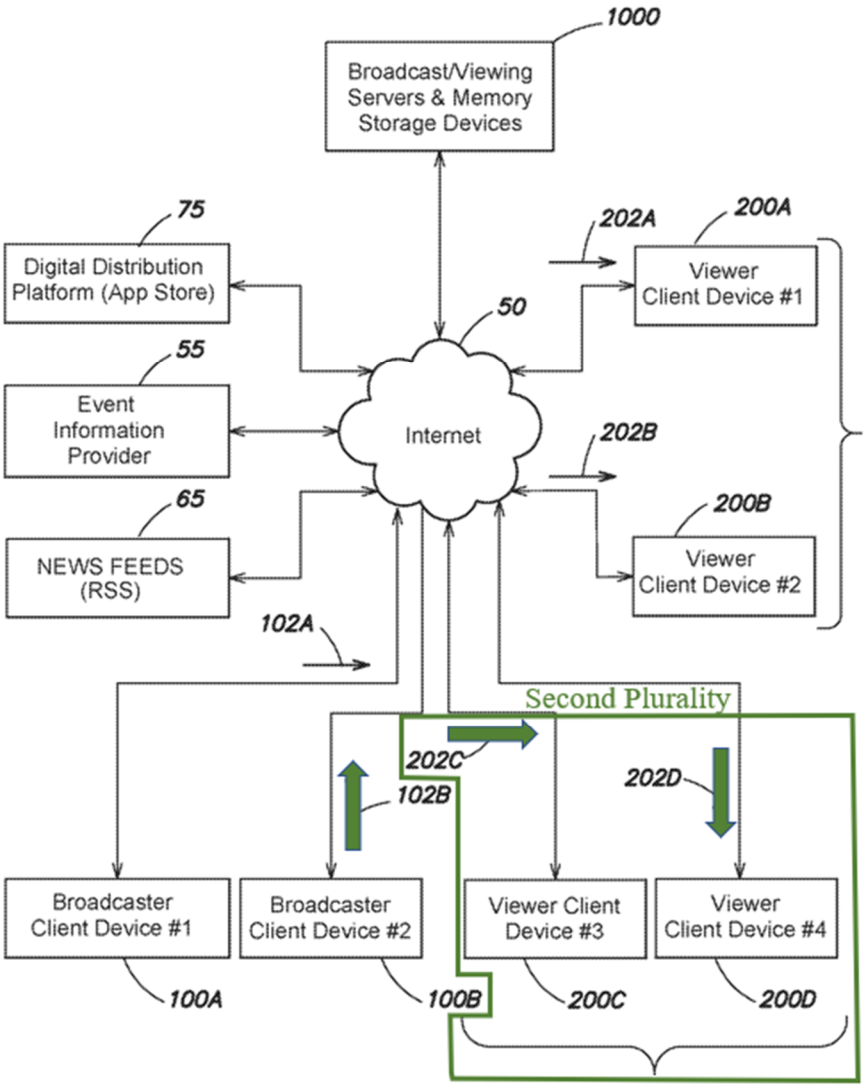
169. In my opinion, Ellis discloses that the content source "may be any suitable source of media content, such as, for example a cable system headend, satellite television distribution facility, television broadcast facility … Internet or network media/web server, or any other suitable facility or system 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, in my opinion Ellis discloses content sources that receive multiple broadcaster's live streams of digital content and distribute that content to user equipment.

iv. **Elements 1.3-1.4: "provide a first copy of the first plurality of copies to a first viewer client device of the first plurality of viewer client devices via a first video Internet communication channel between the plurality of media sources and the first viewer client device of the first plurality of viewer client devices; and" "provide a first copy of the second plurality of copies to a first viewer client device of the second plurality of viewer client devices via a second video Internet communication channel between the plurality of media sources and the first viewer client device of the second plurality of viewer client devices;"**

170. The '697's Figure 1A gives an example of the "first viewer client device [("VCD")] of the first plurality of viewer client devices." There, a first VCD means one user, 200A, of the first group that receives individual copies 202A, 202B of the first live stream of digital content 102A:



Ex.1006, 9:16-41, Fig. 1A (explaining that Figure 1A depicts providing "a first [VCD] (200A)" of "a first plurality of [VCDs] (200A, 200B)" that receive "copies (202A, 202B) of a first broadcaster's live stream of digital content (102A)."). Likewise, "the first viewer client device of the second plurality of viewer client devices" simply means one user of the second group of VCDs that receives individual copies of a second live stream of digital content:

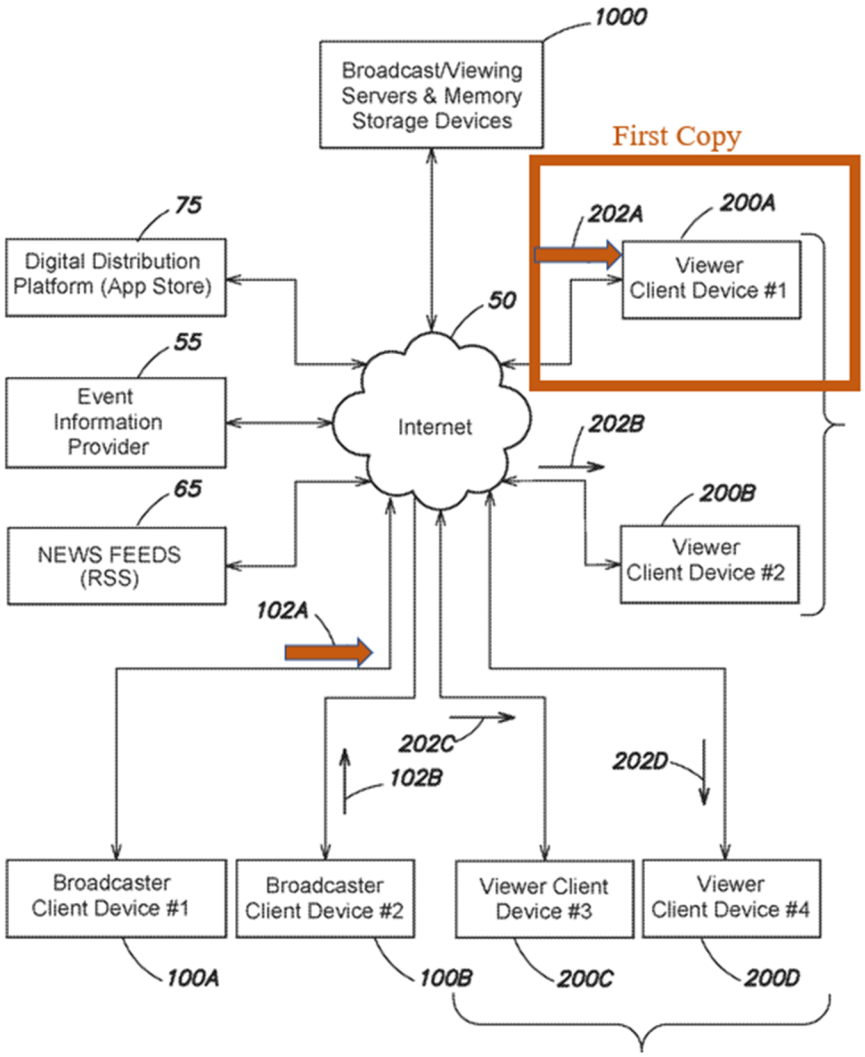


Id., 9:16-41 (explaining that "a first [VCD] (200C)" of "a second plurality of [VCDs] (200C, 200D)" receive "copies (202C, 202D) of a second broadcaster's live stream of digital content (102B)."), Fig. 1A.

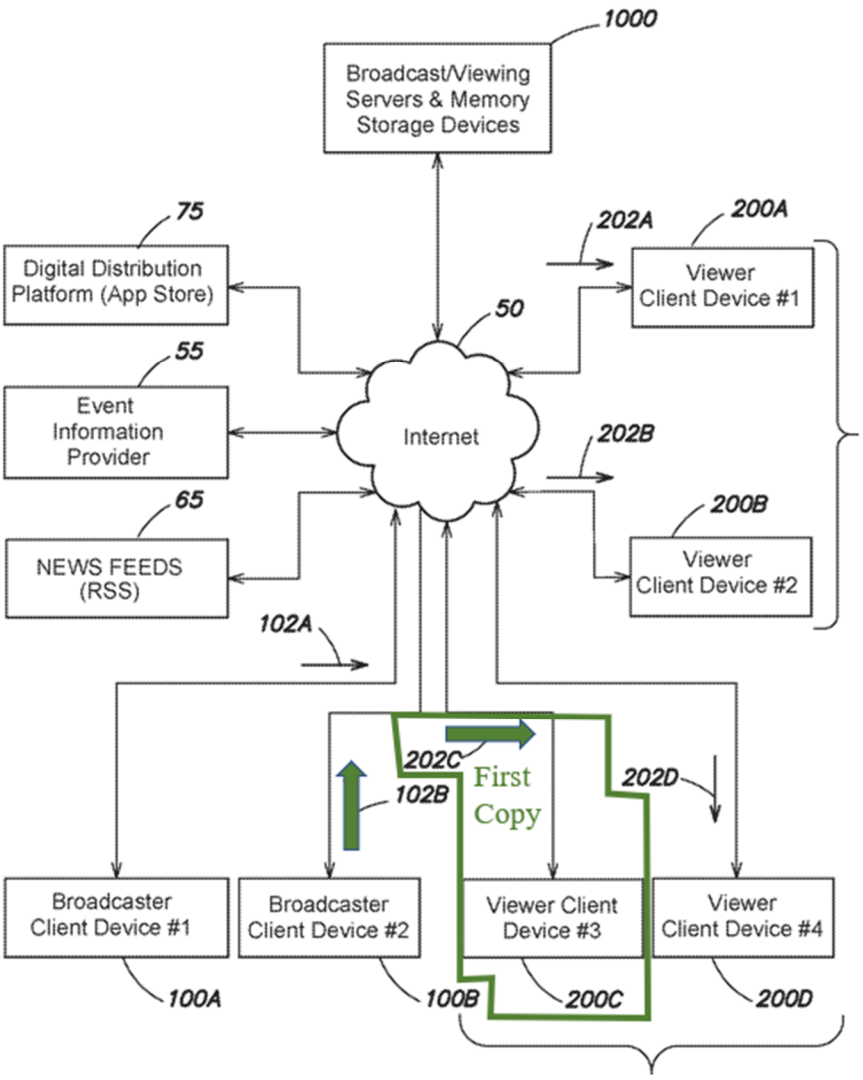
171. In my opinion, Ellis discloses a first VCD of a first/second plurality of VCDs because Ellis discloses an "interactive media system" where "any suitable number of users" may "select any game from the listing" of "available games" to "watch the selected game." Ex.1016 ¶94. For example, a first user in a first group

of users (the first plurality) can choose to watch the Yankees broadcast from the YES network. *Id.* ¶55. Meanwhile, a first user of a second group 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. Thus, it is my opinion that Ellis discloses multiple first users of multiple different pluralities of VCDs.

172. Further, as shown in Figure 1 below, the '697 explains that the first copy of "the first plurality of copies" means a single copy provided to one VCD, where another VCD receives a separate copy over a separate connection:



Ex.1006, 9:16-41 (explaining that VCD 200A receives a "first copy (202A)" of "a first plurality of copies (202A, 202B) of a first broadcaster's live stream of digital content (102A)"), Fig. 1. Likewise, the first copy of "the second plurality of copies" means a single copy provided to one VCD, where another VCD receives a separate copy over a separate connection:



Id., 9:16-10:3 (noting that the system provides "a first copy (202C)" of "a second plurality of copies (202C, 202D) of a second broadcaster's live stream of digital content (102B)."), Fig. 1.

173. In my opinion, 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. Ellis's content source, an internet server, provides the live stream over a connection path, which can be any "suitable wired or wireless path" including an internet connection. Ex.1016 ¶¶39, 45. Thus, it is my opinion that each user would receive an individual copy of the live stream of digital content.

174. In my opinion, Ellis additionally discloses providing individual copies of multiple different live streams to multiple groups of client devices. Ellis further discloses providing digital content or video content over a first/second video Internet communication channel. As an example of an "Internet communication channel," the '697 describes a "video" channel between a media server and a client device to convey video-based commentary. Ex.1006, 5:36-49. Ellis discloses providing video from a content source to user equipment, over a communication path. Ex.1016 ¶39, 74. The content source can be an "internet or network media/web server" and the user equipment "may include computer equipment." *Id.* ¶¶40, 48. The communication path can be "for example, a satellite path, a fiber-optic path, a cable path, or any other suitable wired or wireless path." *Id.* ¶45. 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, in my opinion

Ellis discloses that each user receives a copy from the content source over a video internet communication channel/path that is unique to each user device. *Id*.

v. **Element 1.5: "B) a control server to periodically retrieve, via the Internet and from an event information provider, first event information germane to a first live sporting event, wherein the first event information includes first score information for the first live sporting event; and"**

175. In my opinion, Ellis and Spivey both disclose this element. The '697 explains that the control server is a generic server. Ex.1006, 55:60-6:3. The control server retrieves "event information" from an "event information provider." *Id.*, 21:37-42. The control server then transmits the event information to the socket server. *Id.*, 19:41-45, 21:37-48. The event information is "germane to" a particular "live sporting event," *e.g.*, score information of one of the broadcaster's live streams of digital content. *Id.*, 19:27-37, 21:37-42.

176. In my opinion, 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." Ex.1016 ¶¶41, 42, 142. The SCS periodically retrieves, via the internet and from an event information provider, first event

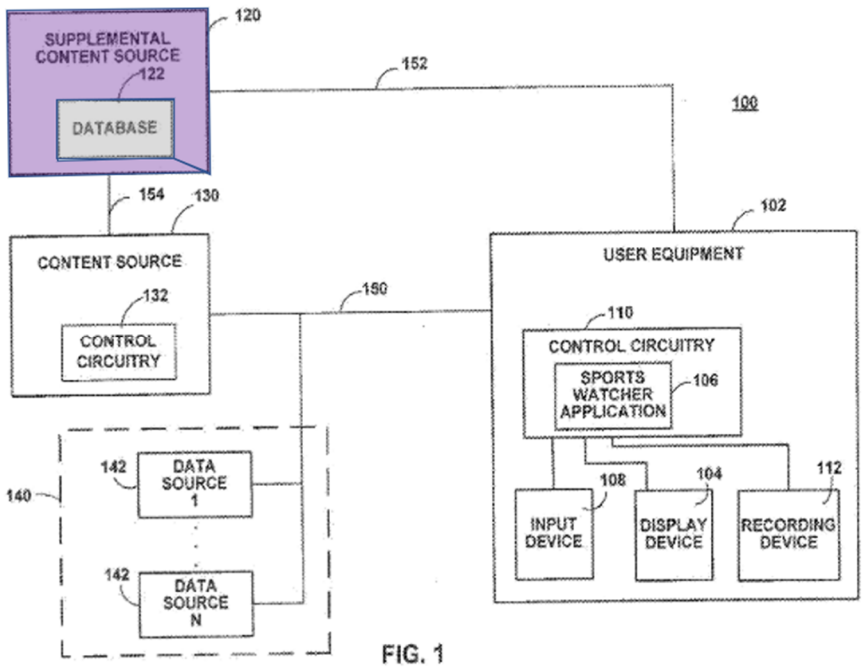
information germane to the first/second live sporting events, including first/second score information for the first live sporting event because the SCS "provide[s] real-time sports scores and game summary information, … spread and wagering data, and any other sports-related information to user equipment 102." *Id.* ¶41; *see id.* ¶¶72, 117. In my opinion, a POSA would have understood that this information is retrieved via the internet. *Id*. ¶¶37-38, 40, 142. Furthermore, a POSA would have understood that this was how real-time data was routinely transferred to users at the time of Ellis and Spivey—including score updates and wagering information. Ex.1016 ¶¶42, 142. Additionally, in my opinion, 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. Ex. 1024, 408-410.

177. While a periodic time interval is not mentioned in Ellis, it is my opinion that 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. This is especially true given that no specific time interval is specified by the claim.

178. 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, yet to limit queries to the system and load on the components. Too many requests in a small

period of time may overload servers, decreasing responsiveness and without providing any real benefit. In my opinion, a POSA would have recognized the period of retrieval should be tailored to the nature of the event and/or the use case. For example, polling for scoring updates on a basketball game at a frequency of 1000 times a second may provide little benefit where the frequency of scoring is tied closely to a 24-second shot clock.

179. 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. To the extent Ellis does not expressly disclose or render obvious this element, it is my opinion that Spivey's live data server device

("LDSD"), as shown in Figure 4, contains the same basic components as the '697's control server:

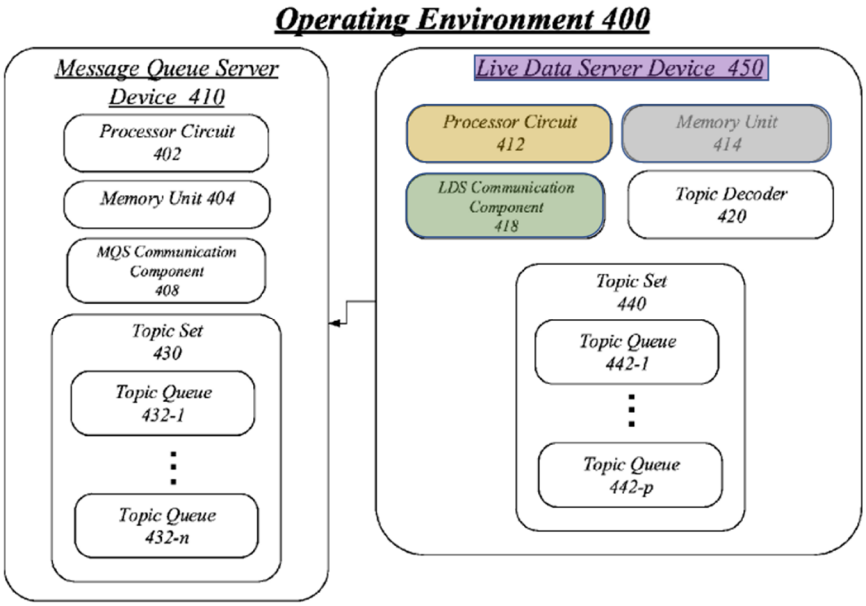


FIG. 4

Ex.1017, ¶¶54-55, Fig. 4 (annotated).

180. The LDSD is a generic server comprising one or more processors, memory devices, and communication interfaces. *Id.* Moreover, it is my opinion that the LDSD performs the same functions as the '697's control server. The LDSD contains an LDS communication component that allows LDSD "to receive live data and transmit the data payloads to MQS device 410." Ex.1017 ¶55. The live data received by the LDSD includes "sports scores and breaking news" and when related to baseball can include "a score, a play, an error, an injury, a player-at-bat, and so forth." *See id.* ¶¶27, 48, 66 ("A data payload may include … information about the live event in progress, such as a score, a name of the player that scored or made a

play, an injury, a penalty, an error, a new record set, and so forth."). The LDSD retrieves the first event information germane to a first/second live sporting event including the first/second score information for the first/second 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. Spivey also discloses a "Memory Unit 414" for storing the event information. Thus, it is my opinion that a POSA would have understood that Ellis and Spivey both disclose this element. To the extent that Ellis's SCS did not meet this claim element, I believe that a POSA would have been motivated to substitute Ellis's SCS with Spivey's LDSD and MQSD for several reasons, including to decrease latency, and a POSA would have had a reasonable expectation of success because this would have involved nothing more than a simple substitution/combination of well-known, compatible elements to achieve a predictable result. §§IX.A *supra.*

vi. **Element 1.6: "C) at least one socket server communicatively coupled to the control server to:"**

181. In my opinion, Ellis in view of Spivey discloses a socket server communicatively coupled to the control server (which, as discussed above, is disclosed by Ellis's SCS and Spivey's LDSD). The '697 describes the socket server as a generic server. *See* Ex.1006, 55:60-6:3. The purpose of the socket server is to

"facilitate communication" of "synchronized event information retrieved by the control server" associated with an event between the control server and the VCDs. *Id.*, 19:64-20:5. 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 the event information in "event messages" via the event information channel "in a synchronized manner" to all viewers "following the particular event." *Id.*

182. Ellis discloses providing event information over a separate connection to user equipment. Ex.1016 ¶¶39-41, Fig. 1. Ellis also discloses a system where multiple servers may provide the event information. Ex.1016 ¶47 (noting the SCS's functionality "may be … provided by multiple systems at multiple facilities").

183. In my opinion, a POSA would have looked to Spivey to add additional backend architecture to reduce latency in the real-time data delivery system. Like the '697'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, it is my opinion that a POSA would have understood that Spivey's MQSD carries out the same purpose as the '697's socket server. The MQSD "receive[s]" and "process[es] live data to provide live data" to "the subscribed connected client devices substantially simultaneously." *Id.* ¶¶36; *see id.* ¶¶45

("[MQS] device 310 may be a representative example of message queue server 120"), ¶54 (describing that 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:"

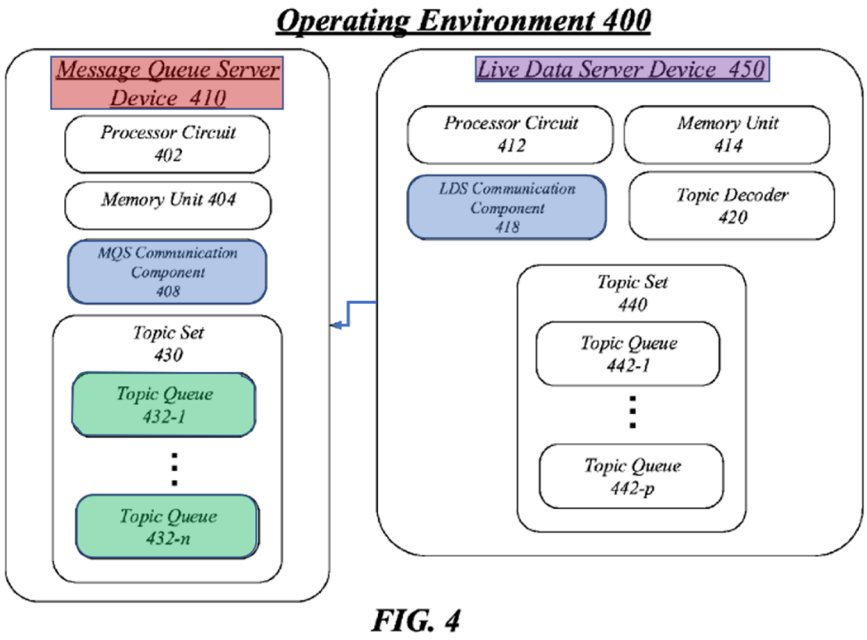


FIG. 4

Ex.1017, ¶¶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 (disclosing that communication component 308 transmits and receives data), ¶54 (disclosing that communication component 408 is analogous to 308), ¶55 ("LDS communication component 418 … allows [LDSD] to … transmit the data payloads to [MQSD]."). Then, 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. After sorting the data into topic queues, the MQSD shares the information "substantially simultaneously" to "the subscribed connected client devices." *See id.* ¶96. Thus, it is my opinion that a POSA would have understood Spivey's MQSD discloses the '697's socket server being communicatively coupled to a control server.

184. Alternatively, Spivey teaches that latency associated with the broadcast of event data, such as sports scores, to clients can be reduced by "using topic cues and open duplex communication connections." *Id.* ¶¶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."). As an example of such "open duplex communication connections" Spivey discloses that communications between a server and a client 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"); *see also*, *Id.* ¶¶ 43, 61, 74, and 92; claim 8. Thus, it would also have been obvious to a POSA in view of these teachings of Spivey 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. In my opinion, a POSA would have recognized that doing so would further the express goals of Spivey with respect to reduction of latency in such

transmissions. In my opinion, a POSA would have had a reasonable expectation of success implementing this web socket functionality in the MQSD because this would have involved nothing more than a combination of well-known, compatible elements to achieve a predictable result.

vii. **Element 1.7: "receive from the control server at least the first score information; and"**

185. In my opinion, Ellis in view of Spivey discloses this element. The LDSD (the control server) "receive[s] live data and transmit[s] the data payloads to [MQSD (the socket server)]." Ex.1017 ¶¶55. Data payloads may include "a score, a player name, a play, and so forth …."

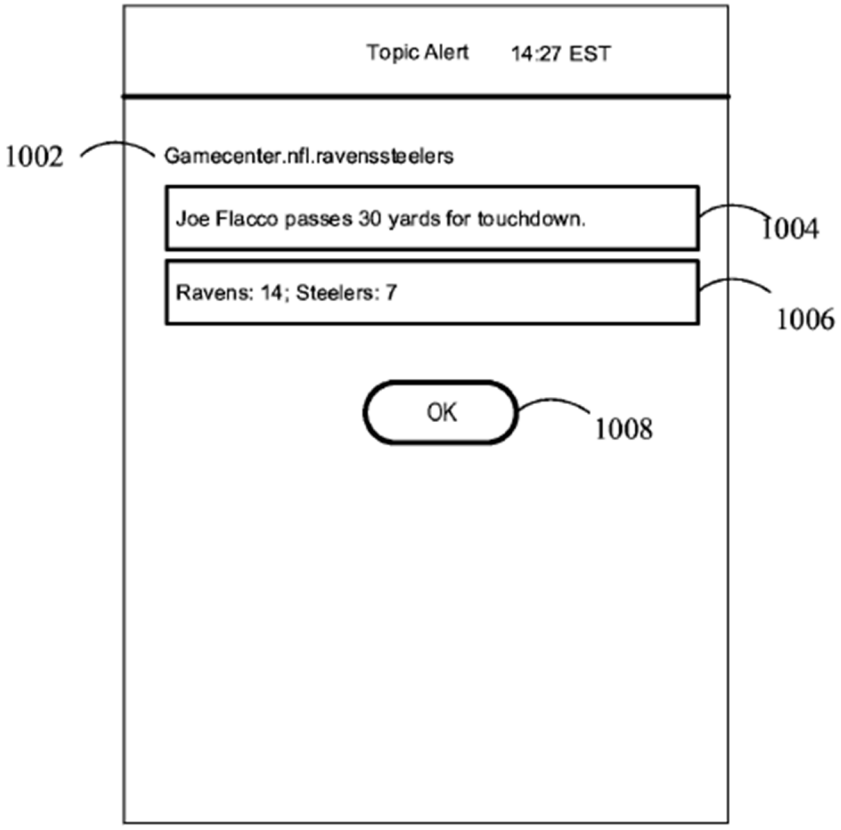


FIG. 10

Id. ¶85, 117, Figs. 9-10. Thus, it is my opinion that a POSA would have understood that Spivey discloses a socket server receiving from the control server at least the first score information. See §§IX.A.2, X.A.1.a.vi; *supra*.

viii. **Element 1.8: "transmit the first score information to at least the first viewer client device of the first plurality of viewer client devices 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 viewer client device of the first plurality of viewer devices, wherein the at least one first event socket corresponds to the first event information germane to the first live sporting event."**

186. In my opinion, Ellis in view of Spivey discloses this element. This element requires (1) a first socket on the socket server that is specifically dedicated to serving information about the first live event, (2) a first viewer client device of a group of VCDs that are watching a live stream of the first live event, and (3) transmitting event data about the first live event from the first socket to the VCD using an internet connection. *See* Ex.1006, Claim 1. These features are shown in Figure 2:

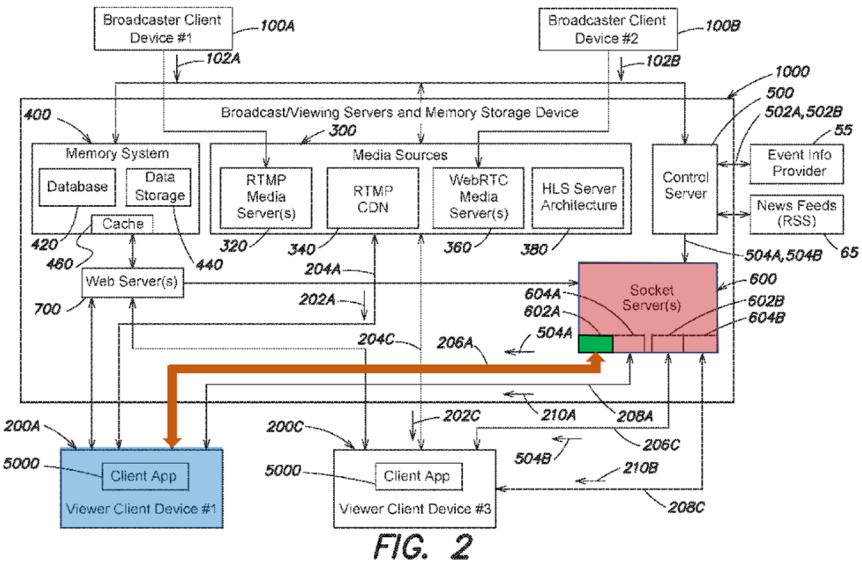


FIG. 2

Id., Fig. 2.

187. Initially, I would note that Spivey often discusses a connection between the client device and the system using a live data access server ("LDAS"), however, Spivey also discloses and teaches that the message queue server device may be "accessible to clients over the network directly." Ex.1017 ¶¶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. Also, the system does not require an LDAS, only that it "may include a[n] [LDAS]." *Id.* ¶38. Thus, it is my opinion that a POSA would have understood Spivey to disclose and teach a direct connection between the client devices and MQSD. In my opinion, a POSA would have used this implementation because reducing the amount of server transfers would reduce latency, which is Spivey's main purpose. Furthermore, it was a well-known alternative to a POSA to implement the functionality of multiple servers as a single server. Ex.1016, ¶47.

188. In my opinion, a POSA would have understood that 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. *Id.* It is my opinion that 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. *See* Ex.1006 at 20:45-61, 21:53-22:6); Ex.1029, pp. 14-20; Ex1030; §IV.C.3., *supra*. In the same way, 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 then 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.

189. 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. *See 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:

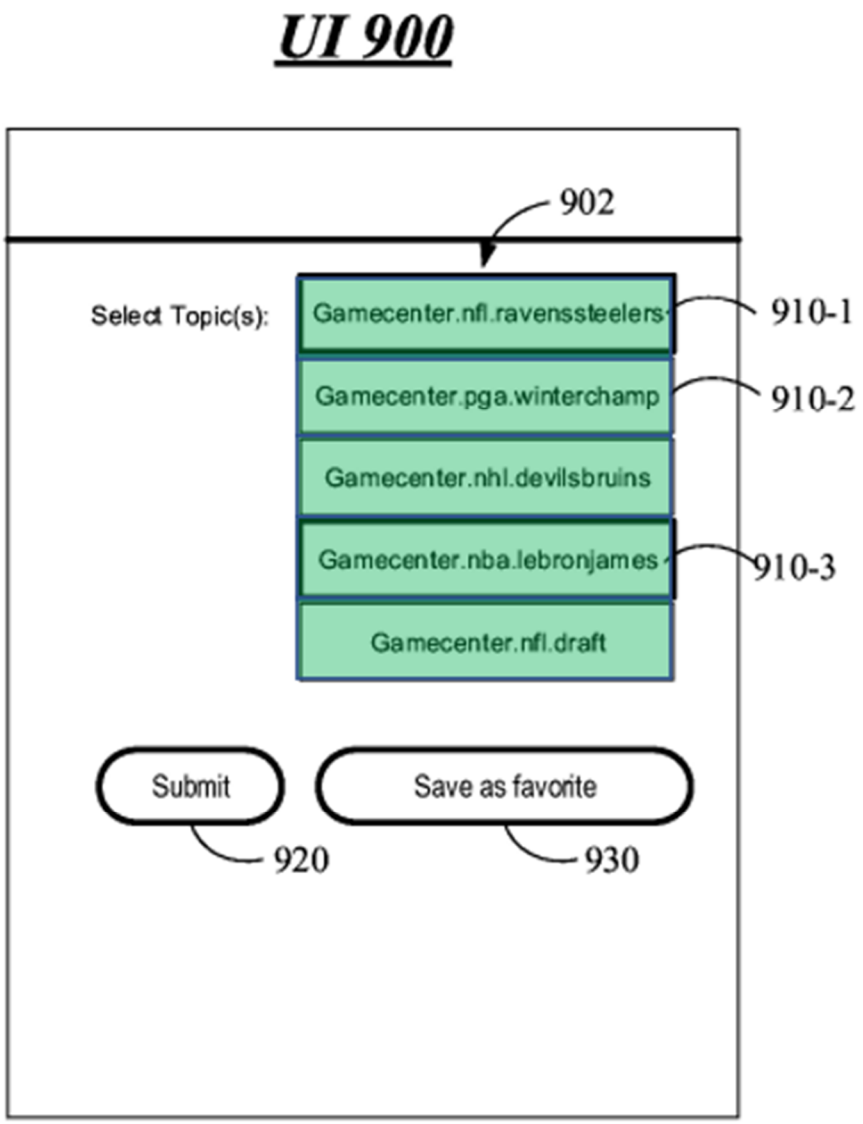


FIG. 9

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, it is my opinion that a POSA would have understood that Spivey's topic queues disclose the '697's first event sockets corresponding to the first event information germane to the first live sporting event.

190. Alternatively, it would also have been obvious to a POSA in view of these teachings of Spivey 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 in §IX.A.

191. In my opinion, Ellis further discloses a first VCD of the first plurality of VCDs. *See* §X.A.1.a.i. Both Ellis and Spivey further disclose transmitting the first score 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. In Ellis, SCS may be configured "to provide real-time sports scores and game summary information, … to user equipment 102." Ex.1016 ¶41. Thus, it is my opinion that Ellis and Spivey disclose the above element.

192. In my opinion, Ellis in view of Spivey additionally 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 Internet." *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, it is my opinion that a POSA would have understood Ellis in view of Spivey to disclose transmitting first event information via a separate internet connection.

193. Further, it is my opinion that this information is provided via a connection between a topic queue and the first user equipment device. Users can access the MQSD using a URL. *See* Ex.1017 ¶¶81-82. Then, the user can access the specific topic queue she wishes to receive information from by using a text string, *e.g.*, a topic label or Event ID, associated with a topic queue. *Id.* ¶179, Fig. 9 ("Gamecenter.nfl.ravenssteelers"). Thus, it is my opinion that 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. Further, I believe 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.

2. '697 Ground 2: Claim 10 is Obvious Over Ellis in View of Spivey and Kellicker

194. For the reasons stated in §IX.B, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, and Kellicker to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 10

i. **Elements 10.0-10.2: "The system of claim 1, wherein: the plurality of media sources comprises: at least one of: at least one real-time messaging protocol (RTMP) media server to receive the first broadcaster's live stream of digital content; and at least one web real time communication (WebRTC) media server to receive the first broadcaster's live stream of digital content; and"**

195. In my opinion, Ellis in view of Spivey and Kellicker teaches these elements.

196. As explained in §IX.B above, Kellicker discloses a server (120/1250) that receives incoming live media streams (e.g., live incoming media stream (1220)), processes the incoming media streams and serves such media streams to playback devices in real-time. Ex.1049, ¶¶100-101, 106. Kellicker discloses that this server can be implemented through a plurality of such 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"), ¶24. Thus, Kellicker discloses a plurality of media sources in the form of real-time streaming media servers.

197. Kellicker also teaches that the server(s) may support various streaming protocols including, RTMP, and WebRTC. *Id*., ¶106. ("computing device 1250 may also support real time messaging protocol (***RTMP***) (and variants thereof), real-time

streaming protocol (RTSP), real-time transport protocol (RTTP), web real-time communication (***WebRTC***), and MPEG-2 transport stream (MPEG-TS)"), ¶ 24 (disclosing transcoders/transmuxers for transforming media streams). Thus, Kellicker discloses media sources in the form of a real-time RTMP media server and/or a real-time WebRTC server.

198. Kellicker teaches providing its server(s) with the capability to "enable the computing device 1250 to process data in accordance with multiple coding technologies and protocols." *Id*., ¶103-104. In my opinion, it would have been obvious to a POSA to have implemented the content source(s) 130 of Ellis in the form of a real-time RTMP media server and/or a real-time WebRTC server. In my opinion, 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.

199. To the extent required by the claims, in my opinion it would have also been obvious to a POSA to have modified Ellis to implement the content source(s) 130 in the form of multiple servers, including a RTMP server ***and*** a WebRTC server. A POSA would have recognized that the use of multiple servers, as taught by Kellicker, would have provided the system with even more flexibility, capacity and adaptability to handle multiple different types of incoming media and comply with multiple types of protocols. *Id*., ¶98.

ii. **Element 10.3: "an HTTP live streaming (HLS) caching server architecture communicatively coupled to at least one of the at least one RTMP media server and the at least one WebRTC media server; and"**

200. In my opinion, Ellis in view of Spivey and Kellicker teaches this element.

201. Kellicker teaches that the at least one RTMP media server and/or the at least one WebRTC media server are communicatively coupled to one or more other servers 1280. Ex.1049, ¶102. For example, 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., an edge server), etc. *Id*.; FIG. 12. Kellicker also explains that the data transmitted to the servers 1280 includes "media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof)." *Id*.

202. Kellicker further discloses that the media streams received by the one or more other servers 1280 "may be associated with the same encoding format and transmission protocol or may be associated with different encoding formats and transmission protocols." *Id*. and ¶24 (disclosing transcoders/transmuxers for transforming media streams). Kellicker also specifically teaches providing the real-time streaming server(s) with the HLS streaming protocol ability. *Id*. at ¶¶105-106.

In my opinion, it would have been obvious to a POSA to implement such HLS conversion capability in the form of the one or more other servers 1280, such as media/stream relay server 1281, for Ellis' content source(s). A POSA would have been motivated to provide HTTP live streaming (HLS) media streams, because as was well-known to a POSA, HLS "provides a reliable, cost-effective means of delivering continuous and long-form video over the Internet." Ex.1052, 2 (§1).

203. 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." *Id*. at ¶102. In my opinion, a POSA would have understood a CDN to be a caching architecture. Ex.1052, 2 (§1) ("[HLS] offers compatibility with large-scale HTTP caching infrastructure to support delivery to large audiences"); Ex.1055 Lepeska, 7:19-23 ("[a] content delivery network or content distribution network (CDN) is traditionally known as a system of computer containing copies of data placed at various nodes of a network, such that access to the cached data is improved by increasing bandwidth and reducing access latency"). To a POSA, the fundamental purpose of a CDN is to provide caching capability in order to improve responsiveness to client device requests for served data (in this case streamed media). *See also,* Phillips (Ex.1050). Thus, in my opinion, a POSA would have understood that Kellicker teaches an HLS caching server architecture

communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server.

iii. **Element 10.4: "the system establishes the first video Internet communication channel to provide the first copy of the first plurality of copies of the first broadcaster's live stream of digital content to the first viewer client device of the first plurality of viewer devices between the first viewer client device and the HLS caching server architecture."**

204. In my opinion, Ellis in view of Spivey and Kellicker teaches this element.

205. As can be seen below, Kellicker teaches communicating media streams between the HLS enabled other server(s) 1280 and client devices 1270:

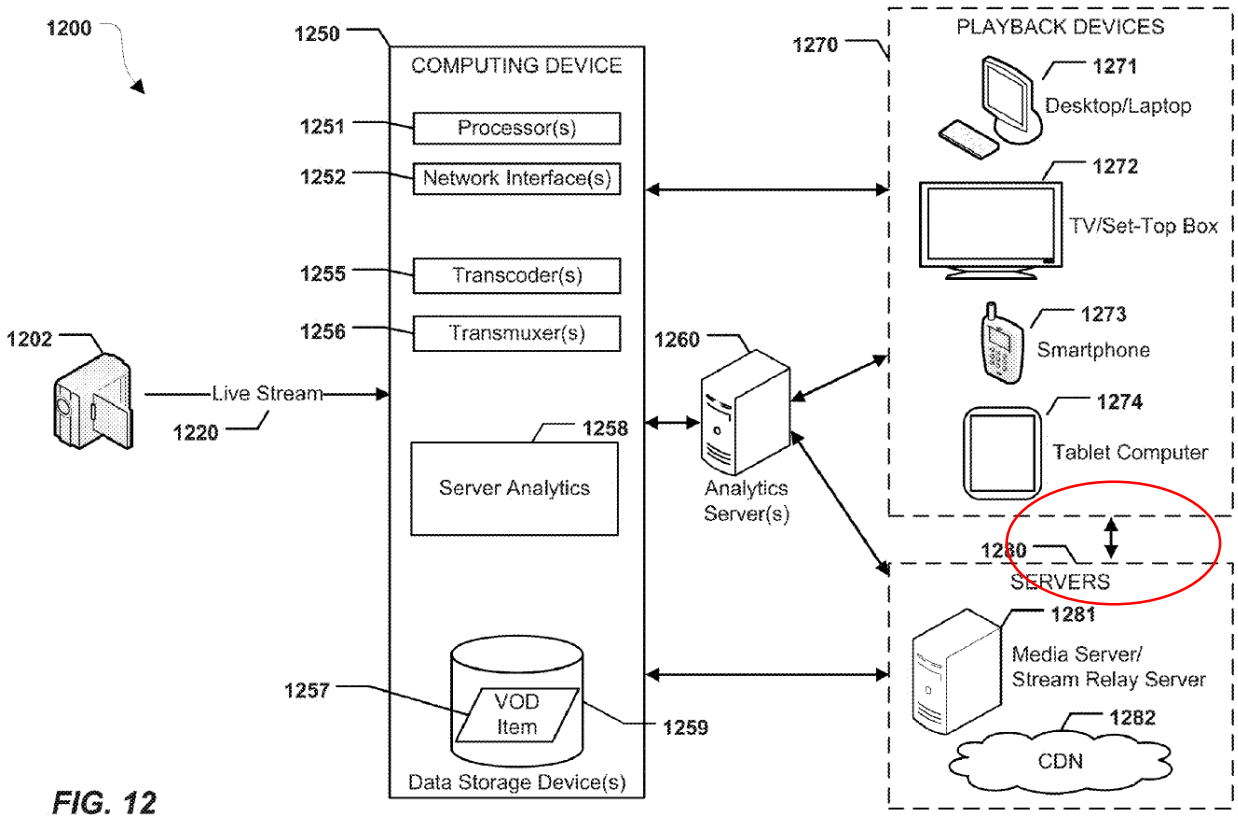


FIG. 12

Ex.1049, Fig. 12 (annotated) (illustrating an Internet communication channel between servers 1280 and playback/client devices 1270), ¶¶19, 98, 111, 102 ("data transmitted to the playback devices 1270 and to the servers 1280 includes media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof").

206. Likewise, Ellis teaches establishing a similar Internet video communication channel 160 between its contents source 130 and user equipment 102. See §X.A.1.a.i-iv.

3. '697 Ground 3: Claims 11 and 15-16 are Obvious Over Ellis in View of Spivey, Kellicker, and Phillips

207. For the reasons stated in §IX.C, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker and Phillips to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 11

i. **Elements 11.0-11.2: "The system of claim 10, wherein the HLS caching server architecture comprises: an HLS mother server communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server to receive, from the at least one of the at least one RTMP media server and the at least one WebRTC media server, a first HLS file suite representing the first broadcaster's live stream of digital content; and a plurality of HLS child servers communicatively coupled to the HLS mother server and an HLS content delivery network (HLS CDN),"**

208. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches these elements.

209. As explained above at §X.A.2., Kellicker teaches an HTTP live streaming (HLS) caching server architecture including a CDN communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server. It would have been obvious to a POSA to have provided this HLS caching server CDN 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." Ex.1050, ¶31. Consistent with the teachings of Phillips, a POSA would have been motivated to provide this server architecture in order to provide "for high-performance streaming . . . to subscribers using one or more Internet-based infrastructures." *Id*., ¶30. For example, Phillips teaches that 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 implement Phillip's 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 . . .").

210. 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, coupled to an HLS CDN architecture (1282), and from the CDN to a client/user device (1270):

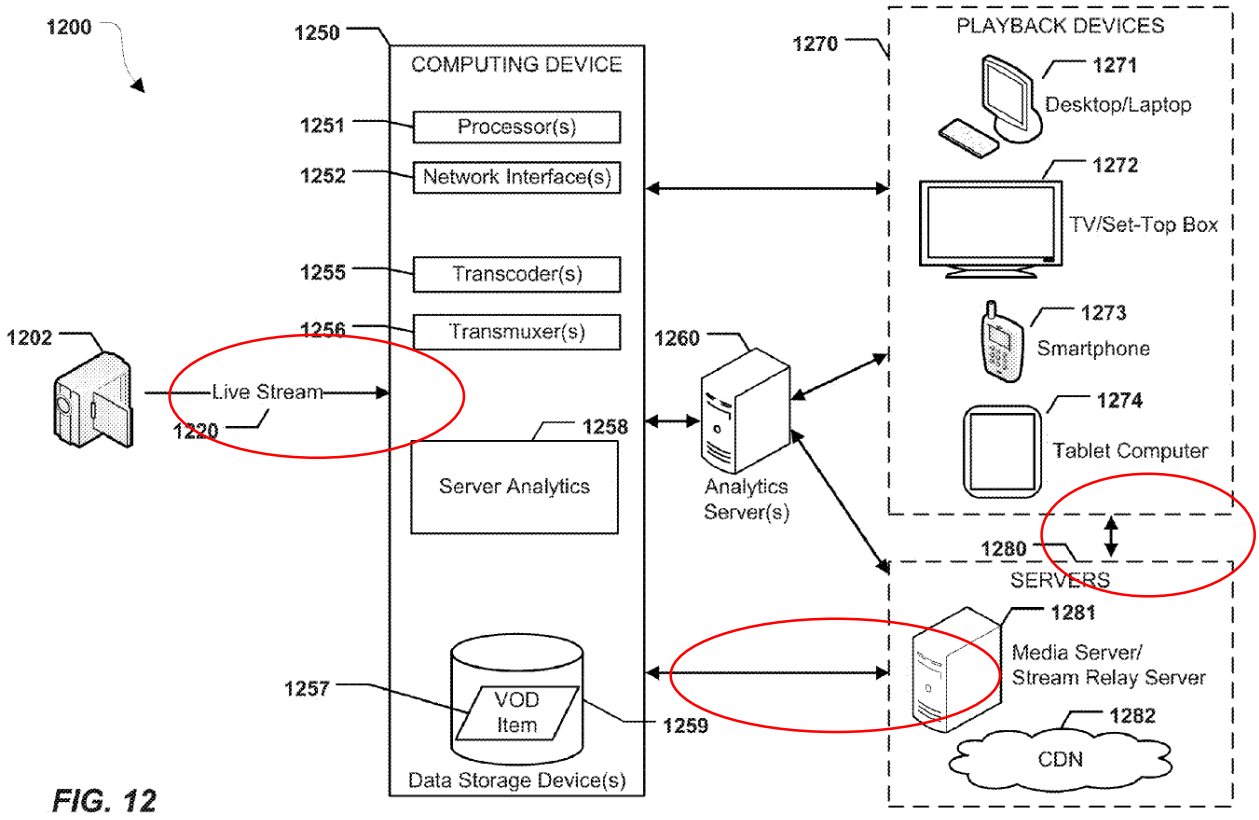


FIG. 12

Ex.1049, Fig.12 (annotated).

211. In particular, Kellicker discloses that the media server(s) 1250 is configured to transmit data to "a server of a content distribution network (CDN) 1282." *Id.*, ¶102. Further, Kellicker discloses that, "[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.

212. In similar fashion, Phillips teaches that content that is 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 that 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 contents and/or nature of the files being handled by the system could be 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.1050, §§8.4-8.6. As can be seen, Phillips’ CDN architecture discloses mother servers (e.g., national server, regional servers), which are mother servers coupled 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.

213. With respect to the "HLS file suite" this represents nothing more than conventional HLS file transfer protocol, as acknowledged in the ‘697. Ex.1006, 38:47-55. 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.1050, §§8.1-8.7. By delivering content from Kellicker’s server 1250 to a Phillips-type CDN, the HLS caching server architecture

comprises: an HLS mother server (e.g., national server, regional servers) which are communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server.

214. Hence, 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 POSA would have understood that 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. As explained in the '697, an "HLS file suite" represents nothing more than the contents of a conventional HLS file transfer protocol. Ex.1006, 38:47–55. In my opinion, 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.1052, 27-29 (§ 8.1-8.7).

215. Alternatively, and to the extent that the nature or content of the data could be considered as limiting the claimed system, 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.1050, §§8.1-8.7. For example, as explained in Kellicker, this HLS caching server architecture receives a portion of the first HLS file suite (video

stream) representing the first live stream of digital content. Ex.1049, ¶102 ("data transmitted to the playback devices 1270 and to the servers 1280 includes media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof"). Further, Kellicker discloses that "first HTTP port(s) may be used to communicate a master rendition list and media chunks . . .," each of which is a portion of the HLS file suite. Ex.1049, ¶110; *see also* Ex.1006, 38:47–55.

216. Similarly, Phillips teaches that the first HLS file suite is received (media chunks and manifest/playlist). For example, Phillips teaches that content that is requested for delivery by the CDN may include a received "manifest" that is made available to a "content delivery node" (e.g., "mother" or "child" server) of the CDN server architecture. Ex.1050, 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"). And, to the extent that 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 (e.g., a "child" server). Ex.1050, Abstract, ¶31, Fig. 1. As another example, Phillips explains that the media portions of the HLS file suite are received by the CDN

architecture. Ex.1050, FIG. 12 ("Pull Missing Segments for ABR Assets", which are media chunks of the HLS file suite), ¶9.

ii. **Element 11.3: "wherein: the system establishes the first video Internet communication channel to provide the first copy of the first plurality of copies of the first broadcaster's live stream of digital content to the first viewer client device of the first plurality of viewer devices between the first viewer client device and the HLS CDN; and"**

217. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches this element.

218. As explained above in §X.A.1.a.iv. 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. As further explained at §§XA.2.a.ii.-iii. and X.A.3.a.i., Ellis as modified by Spivey, Kellicker and Phillips establishes that the first video Internet communication channels are provided between the first viewer client device and the HLS CDN. *See also* Ex.1050, ¶33 ("Each local edge server is operable to serve a plurality of subscribers in a serving area for delivering content to associated end stations or client devices, as illustrated by an example client device or user equipment (UE) device 105 that may for consume or access streaming content delivered via CDN 100 in any type or number of access technologies including broadband access via wired and/or wireless (radio) communications."), ¶28 ("client devices may also access or consume

content/services provided over… packet-switched wide area public network such as the Internet via suitable service provider access networks.").

iii. **Element 11.4: "the first copy of the first plurality of copies of the first broadcaster's live stream of digital content is based on the first HLS file suite."**

219. As explained at §X.A.3.a.i., a POSA would have understood that 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 as required by the HLS specification and the CDN distribution process.

b. Dependent Claim 15

i. **Element 15.0: "The system of claim 11, wherein 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 broadcaster's live stream of digital content and no other HLS file suite representing the first broadcaster's live stream of digital content."**

220. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches these elements.

221. In my opinion the claimed HLS mother/child server architecture is no different than the HLS CDN caching server architecture disclosed and taught by Phillips. Therefore, a POSA would have understood that the structure of Phillips' CDN hierarchy is certainly capable of executing the intended use or functionality recited in claim 15.

222. In my opinion, a POSA would have understood the recitation that 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 broadcaster's live stream of digital content and no other HLS file suite representing the first broadcaster's live stream of digital content*" requires nothing more than the conventional functionality of a conventional caching CDN network.

223. For example, Ex. 1054 - Basile (EP2920708) discusses the purpose of the CDN, which is to avoid burdening the origin server with requests, by caching the content:

> In operation, end user device 120-121 request network content, such as content 135 associated with origin server 130. Instead of these requests being handled by the origin server 140, content node 110 of a content delivery network receives the content requests over ones of links 140-141 and processes the content requests for delivery of the content to the associated end user devices 120-121. Requested network content that is already stored in content node 110 can be provided quickly to the end user devices, while network content that is not already stored in content node 110 can be responsively requested by an associated content node 110 from origin server 130 for caching by content node 110. In this manner, content node 110 can act as an intermediary proxy node to provide local and fast access for end user devices 120-121 to network content of origin server 130 without burdening origin server 130 with many end user content requests.
> *Id*. ¶16.

224. Consistent with this, Phillips discloses a national server 102 of its CDN network that is an exemplary mother server. Ex.1050, ¶¶31, 32, 56, Fig. 12. In my

opinion, a POSA would have understood that the national/mother server of Phillips can act as the primary server where the original/complete version of the served content can be stored, acting as a source for content distributed by the Content Delivery Network (CDN) which can then distribute locally cached copies of that content from geographically distributed edge servers closer to users for faster access. In other words, a POSA would have understood that the national/mother server of Phillips can act as a backend "mother" server that holds the master copies of content that is served through the CDN network. Ex.1050, ¶61 ("If the missing audio, video, and/or subtitle asset segments are within the historical delivery trends/thresholds, such segments may be pulled from a regional or national server node or another edge server node (block 1072).").

225. Additionally, as explained in Phillips, portions of the HLS file suite can come from anywhere in the CDN. Ex.1050, ¶9 ("if any segment files referenced by the manifest are absent from the database…. pull from another content delivery node of the CDN only the representations of the absent segment files satisfying the one or more delivery rules and store the pulled representations in the database cache."), ¶8 ("After determining the compliant representations of 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").

226. Such a design avoids retrieving from the media servers any portions of the HLS file suite that exist anywhere in the CDN. Such operation is disclosed in Phillips. Ex.1050, ¶60 ("When all the loops have been performed for each adaptive streaming manifest in cache, for each child manifest in the master manifest, and for each URL in each child manifest, all segment files should exist in the edge node's local cache memory. Upon verification that all segment files for all manifests exist in the edge node's local cache memory at block 1034, the process moves to block 1036 where the edge node reports to its parent node that defragmentation is complete."). Therefore, because all the segments exist somewhere in the CDN (e.g., at an edge node), no other HLS file suite representing the first live stream of digital content need be requested from the media server.

227. Based on at least the above, in my opinion, it would have been obvious to a POSA, in view of Phillips, to have used the national/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 in a CDN. A POSA would have understood that this implementation allows data being served throughout the network to be replicated throughout and served by the CDN, which reduces load on the content creator server structure, and facilitates the defragmentation objective taught by Phillips.

c. Dependent Claim 16

i. **Element 16.0: "The system of claim 15, wherein the HLS mother server maintains a persistent connection to the at least one of the at least one RTMP media server and the 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."**

228. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches these elements.

229. In my opinion, claim 16 is satisfied on the same basis as claim 15 above, because any connection between the HLS mother server and the media sources would "persist" over some unspecified duration of time.

230. In my opinion, based on the '697, a POSA would have understood that live media stream transmissions, including HLS, RTMP and WebRTC protocols, utilize a "persistent" connection between the source of the media and the server(s) distributing it to a user/client:

> 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.1006, 8:37-45.

231. But, according to the '697, "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. 1006, 33:49-53. However, the connection of claim 16 does not refer to a connection between a server and a client device. Instead, claim 16 refers to the connection between the media source and the HLS CDN mother server.

232. Thus, as would be readily understood by a POSA, 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 utilizes a "persistent connection." In my opinion, a POSA would have understood that the "persistent connection" of claim 16 only need be persistent for "a portion of a duration of the first broadcaster's live stream of digital content."

233. Also, given the nature of the connection between the RTMP media server, the WebRTC media server, and the HLS mother server there would be very little reason to use a non-persistent connection, like an HTTP (request, response, close) connection, where a persistent connection would be more efficient and have lower overhead. For example, a POSA would have known that the creation and closing of connections requires a significant overhead via the allocation/deallocation of memory, use of additional system calls to the operating system, as well as the network overhead. While keeping the connection open prevents the reallocation of certain resources, this is acceptable given the nature of the connection between the CDN/mother server and the media servers as a streaming connection, that does not

need to follow the HTTP protocol. In any event, for a POSA the choice between a "persistent" and "non-persistent" connection would have been nothing more than a typical tradeoff – implicating a design choice of increased overhead and potentially increased resource sharing vs. reduced overhead and reduced resource sharing.

234. Additionally, because the choice of connection is either persistent or non-persistent and both types of connections were well-known in the art, the choice of a persistent connect would have represented a mere design choice for a POSA.

235. 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.

236. In my opinion, a POSA would have appreciated that the operation of claim 16 is consistent with live streaming itself. For example, if all elements of the first HLS file suite were to be received by the HLS mother server after the completion of the live stream, this would merely be more akin to playing a recorded video (VOD). Because Ellis in view of Kellicker and Phillips supports live streaming (as well as VOD) the HLS mother server (e.g., server 1280/1282, national server 102) would maintain a connection to the at least one of the at least one RTMP media server and the 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. Also, this condition will always be satisfied by a persistent connection that persists for the entirety of the broadcaster's live stream.

4. '697 Ground 4: Claims 12-14 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips, and Stoica

237. For the reasons stated in §IX.D, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker, Phillips and Stoica to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 12

i. **Elements 12.0-12.1: "The system of claim 11, further comprising: at least one webserver to provide an applications programming interface (API) between the system and each of the first broadcaster client device, the second broadcaster client device, the first plurality of viewer client devices and the second plurality of viewer client devices;"**

238. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

239. Claim 12 specifies that a webserver provides an applications programming interface (API) between it and a number of devices of the system. However, the nature of the claimed API is unspecified. And, the claimed API has no further connection to claim 12 or any additional dependent claim. In my opinion, a

POSA would have understood that a webserver may provide any of its APIs to any system communicatively coupled to it, and, therefore, the webservers taught by Ellis in view of Spivey, Kellicker, and Phillips (as discussed above) disclose this claim element.

240. Also, Stoica expressly teaches at least one webserver that provides an API. For example:

> 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 (emphasis added).

241. Also, Stoica teaches a webserver in the form of a "dispatcher server" (1114) that may be 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.

242. Stoica's gateway 1122 and dispatcher server 1114 are each communicatively coupled with both client devices and the streaming content servers (e.g., CDN and origin servers). *Id.*, 9: 63–10:2; 11:17–32, 2:49–53; Fig. 11. As previously discussed, content requested by client devices, including first and second viewer content devices, is provided via a CDN, according to the system architecture

taught by previous combinations of cited references, which included communicatively coupled first and second pluralities of broadcast client devices providing streaming media content for distribution by the CDN. *See* §§VIII.A-C, X.A.1-3.

243. Stoica also teaches that the CMS is a webserver that provides an application 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 ("the CMS provides metadata associated with a video stream, including the information required to deliver the content.").

244. In its interactions between the CDN and client devices, Stoica teaches that the CMS/CDC can provide an augmented URI (i.e., VRL) specifying a manifest file to a client device:

> 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.").

245. It is clear that the CMS/CDC are communicatively coupled with both client devices and the streaming content servers (e.g., CDN and origin servers). *Id.*, Fig. 8, 2:49–53. As previously discussed, content requested by client devices, including first and second viewer content devices, is provided via a CDN, according to the system architecture taught by previous combinations of cited references, which included communicatively coupled first and second pluralities of broadcast client devices providing streaming media content for distribution by the CDN. *See* §§VIII.A-C, X.A.1-3.

ii. **Element 12.2: "a memory system communicatively coupled to the at least one webserver, the memory system including a database storing: first stream information for the first broadcaster's live stream of digital content including the first live sporting event video-based commentary from the first broadcaster client device, wherein the first stream information includes a stream identifier (StreamID) for the first broadcaster's live stream of digital content,"**

246. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

247. For example, Stoica teaches that the at least one of the webservers is communicatively coupled to a memory system including a database. In further detail, 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.

248. Stoica explains that this 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="NewsAtl0") OR (show="SundayMagazine").
> *Id*., 10: 31–49.

249. 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 commentary from the first broadcaster client device because such data by definition includes the stream identifier (StreamID). *See* Claim 12 ("the memory system including a database storing: first stream information for the first broadcaster's live stream of digital content including the first live sporting event video-based commentary from the first broadcaster client device, wherein the first stream information includes a stream identifier (StreamID) for the first broadcaster's

live stream of digital content"). However, it is also clear that Stoica's database stores other information (e.g., viewer data) for the first broadcaster's live stream of digital content including the first live sporting event video-based commentary from the first broadcaster client device.

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

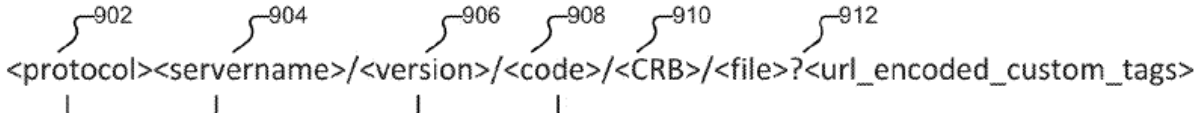
> Gateway 1122 also provides access controls and persistently stores the information regarding the API requests, data access and presentation policies, and filter descriptions. The information is maintained in a persistent database, such as mySQL or Oracle database.
> Ex.1051, 11:28-32.

251. Additionally, Stoica teaches that through the gateway 1112 API, data is available on a per filter basis. *Id*., 11:17-19. In my opinion, based on at least the above, a POSA would have understood that the stream identifier (StreamID) information is utilized for filtering, and ultimately routing content requests and retrieval. In my opinion, a POSA would have also understood that Stoica teaches that gateway 1122 is coupled to a database storing stream identifier (StreamID) information. The StreamID, which is an identifier for a particular source of content, *e.g.*, a "show" field in the case of Stoica.

iii. **Element 12.3: "wherein the at least one webserver: determines an address for the HLS CDN to establish the first video Internet communication channel to provide the first copy of the first plurality of copies to the first viewer client device, based at least in part on the StreamID for the broadcaster's live stream of digital content; and"**

252. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

253. 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 a 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."). Stoica FIG. 9 (excerpted below) depicts an exemplary VRL, which is transferred to client devices by a CMS webserver.



254. Next, the CDC determines the address for the HLS CDN, which is based on the VRL:

> 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 (emphasis added); FIG. 8 (illustrating transfer of video from CDN to client player beginning with a VRL).

255. Stoica further teaches that a number of optional parameters 912 may be embedded in the VRL, one of which is a "show" parameter (e.g., a StreamID). *Id*., 7:25-8:1-4 ("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 (discussed above) stored in the memory database of the RSP component (discussed above), which is communicatively coupled to the webserver portion of the dispatcher server.

iv. **Element 12.4: "transmits the address for the HLS CDN to the first viewer client device of the first plurality of viewer client devices to establish the first video Internet communication channel."**

256. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teach this element.

257. 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. *See e.g.,* Ex.1052, 24 (§6.3.1) ("How the client obtains the URI to the Playlist file is outside the scope of this document; it is presumed to have done so. The client obtains the Playlist file from the URI."):

> [I] nstead 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. . . . 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.

258. Once the client receives the playlist/manifest, the client may request a segment to establish the first video Internet communication channel to receive the media stream. Ex.1052, 24 (§6.3.2) ("Every time a Media Playlist is loaded or reloaded from a Playlist URI, the client MUST determine the next Media Segment to load"), Ex.1051, FIG. 8 (illustrating transfer of VRL, manifest, and a video stream from a CDN to client player/device).

b. Dependent Claim 13

i. **Elements 13.0-13.1: "The system of claim 12, wherein the first stream information for the first broadcaster's live stream of digital content further includes an event identifier (EventID) for the first live sporting event; and"**

259. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches these elements.

260. As explained above in §§X.A.1a.i-iv, Ellis in view of Spivey 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. Ellis in view of Spivey also teaches 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. *See* §§X.A.1a.vi-viii; *See also*, Ex.1017, ¶92.

261. Spivey teaches 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. As can be seen, this topic label (EventID) "relates to" a specific sporting event or "show" ("ravenssteelers"), which is an example of a StreamID.

262. When further combined with the database system of Stoica, it would have been obvious for a POSA to store these EventIDs, in order to filter and store analytics information. Ex.1051, 9:46-10:54 ("Clients 1104-1112 each include a monitoring module that collects status information. As one example, performance information associated with video playback is collected, is other information about the client… the client is mapped to a dispatcher server…dispatcher server (1114) includes two conceptual modules. The first module implements a communication interface for receiving status information from clients… second task performed is to compute snapshots and on-line statistics over the telemetry data matching each filter").

263. In other words, storing stream information such as EventIDs amounts to nothing more than a design choice to transmit information already available to a client as taught by Spivey using mechanisms already provided by Stoica (e.g., dispatcher server and a database).

ii. **Element 13.2-13.3: "the at least one webserver further: determines a first socket address for the at least one first event socket of the at least one socket server based at least in part on the EventID; and transmits the first socket address to the first viewer client device of the first plurality of viewer client devices to establish the first event information Internet communication channel."**

264. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

265. Spivey discloses that its MQSD and/or LDSD may be or include webservers. 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.

266. According to Spivey, "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").

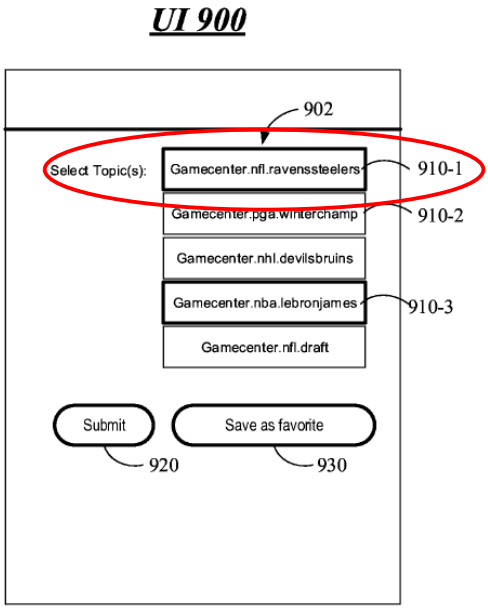


FIG. 9

267. In my opinion, 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." *See also*, §§X.A.1.a.vi-viii. 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.

268. As discussed in §§X.A.1.a.vi-viii above, it would have been obvious to a POSA in view of these teachings of Spivey for the MQSD server, which may be a webserver, to transmit the first socket address to the first viewer client device to establish a communication channel 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. Ex.1017 ¶¶53, ¶35, Fig.4.

269. In addition, as discussed above §§X.A.4.a.1.-iv, Stoica teaches an implementation of a webserver architecture as an intermediary between a client and a CDN. Stoica teaches that doing so facilitates communications and efficiency of the client's access to 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. In my opinion, based on the teachings of Stoica, it would have also been obvious to a POSA, to have transmitted the first socket address to the client to establish the first

event information Internet communication channel via an intermediate web server such as Stoica's CMS or CDC.

c. Dependent Claim 14

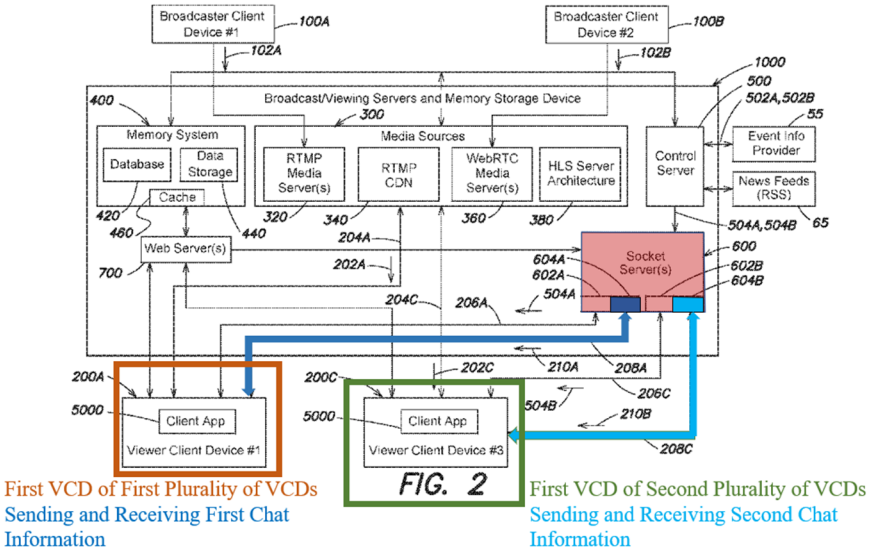
i. **Element 14.0-14.1: "The system of claim 13, wherein in C), the at least one socket server further transmits and receives first chat information regarding the first live sporting event video-based commentary included in the first broadcaster's live stream of digital content via at least one first chat/system event Internet communication channel between at least one first chat/system event socket of the at least one socket server and the first viewer client device of the first plurality of viewer client devices, wherein the at least one first chat/system event socket corresponds to the first chat information regarding the first live sporting event video-based commentary; and"**

270. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

271. As discussed above, 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. §§X.A.1.a.

272. The '697 explains that a chat/system event communication channel may be any generic internet communication channel. Ex.1006, 5:29-36. These elements only further require that the system "transmits and receives" first "chat

information regarding" the first "live sporting event video-based commentary included in the first stream of digital content" over another separate, individual connection between a first "chat/system event socket" and a first VCD:



Ex.1006, 22:7-39, Claim 24, Fig. 2. In sum, it is my opinion that a POSA would have understood that these elements require that chat information is transmitted and received over a communication channel from another socket that is dedicated for chat/system information.

273. Ellis further 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. In my opinion, a POSA would have understood that Ellis discloses a system that transmits and receives first chat information regarding the first live sporting event video-based commentary included in the first broadcaster's live stream of digital content. Ex.1016 ¶126. See IX.D. *supra*.

274. 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.

275. In my opinion, as explained in §§X.A.1a.vi-viii, it would been obvious to a POSA to have established these communication channels/connections with web socket connections between the client and a web socket server.

276. 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. 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.

277. 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 would have been separate from the chat information according to the routing methodology applied by Spivey's socket server. *Id*.

278. In my opinion, based on the considerations above (hierarchy and exact matching) it would have also been obvious to a POSA in view of Spivey to provide a first chat/system event information socket (*e.g.*, "gamecenter.nfl.ravenssteelers.chat") corresponding to the first chat information regarding the first live sporting event video-based commentary.

ii. **Elements 14.2-14.3: "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 based at least in part on the StreamID; and transmits the second socket address to the first viewer client device of the first plurality of viewer client devices to establish the first chat/system event Internet communication channel."**

279. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

280. As discussed above, Spivey teaches that event information such as statistics would have been kept separate from chat information (on a separate channel/socket). 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 element 14.1.

281. In my opinion, in 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 and facilitating the use of hierarchical organization. Ex.1017 ¶¶48, 51-52. As discussed at §X.A.4.a.ii-iii

above, a StreamID as discussed in Stoica would have been a "show." 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").

282. Alternatively, in my opinion using a StreamID rather than an EventID would have been 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.

283. As discussed in §§X.A.1a.vi-viii above it would have been obvious to a POSA in view of these teachings of Spivey for the MQSD server, which may be a webserver, to transmit the first socket address to the first viewer client device to establish a communication channel 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. Ex.1017 ¶¶53, ¶35, Fig.4.

284. In addition, as discussed above §X.A.4.a, Stoica teaches an implementation of a webserver architecture as an intermediary between a client and a CDN. Stoica teaches that doing so facilitates communications and efficiency of the client's access to 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. In my opinion, based on the teachings of Stoica, it would have also been obvious to a POSA, to have transmitted the second socket address to the client to establish the first chat/system event Internet communication channel via an intermediate web server such as Stoica's CMS or CDC.

5. '697 Ground 5: Claims 17-18 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips, and Pantos

285. For the reasons stated in §IX.E, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker, Phillips, and Pantos to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 17

i. **Elements 17.0-17.1: "The system of claim 1, wherein:";**

ii. **"the plurality of media sources comprises: at least one of: at least one real-time messaging protocol (RTMP) media server to receive the first broadcaster's live stream of digital content; and at least one web real time communication (WebRTC) media server to receive the first broadcaster's live stream of digital content; and";**

iii. **"an HTTP live streaming (HLS) caching server architecture communicatively coupled to at least one of the at least one RTMP media server and the at least one WebRTC media server;"; and**

iv. **"the system establishes the first video Internet communication channel to provide the first copy of the first plurality of copies of the first broadcaster's live stream of digital content to the first viewer client device of the first plurality of viewer devices between the first viewer client device and the HLS caching server architecture;"**

286. Elements 17.0-17.1 are substantially the same as elements 10.0–10.4. *See* §X.A.2.

v. **Element 17.2: "the HLS caching server architecture comprises: an HLS mother server communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server to receive, from the at least one of the at least one RTMP media server and the at least one WebRTC media server, a first HLS file suite representing the first broadcaster's live stream of digital content;";**

vi. **"a plurality of HLS child servers communicatively coupled to the HLS mother server and an HLS content delivery network (HLS CDN);";**

vii. **"the system establishes the first video Internet communication channel to provide the first copy of the first plurality of copies of the first broadcaster's live stream of digital content to the first viewer client device of the first plurality of viewer devices between the first viewer client device and the HLS CDN;"; and**

viii. **"the first copy of the first plurality of copies of the first broadcaster's live stream of digital content is based on the first HLS file suite;"**

287. Elements 17.2 are substantially the same as elements 11.1– 11.4. See §X.A.3.a.

ix. **Element 17.3: "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 broadcaster's live stream of digital content and no other HLS file suite representing the first broadcaster's live stream of digital content;**

288. Element 17.3 is substantially the same as element 15.0. See §X.A.3.b

x. **Element 17.4: "the HLS mother server maintains a persistent connection to the at least one of the at least one RTMP media server and the 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;"**

289. Element 17.4 is substantially the same as element 16.0. See §X.A.3.c.

xi. **Element 17.5: "the first HLS file suite provided by the at least one of the at least one RTMP media server and the at least one WebRTC media server to the HLS mother server comprises: a first HLS playlist specifying a first plurality of different resolution versions of the first broadcaster's live stream of digital content, the first HLS playlist further including a corresponding first plurality of chunklist pointers to first respective chunklists for the first plurality of different resolution versions;";**

xii. **"the first respective chunklists for the first plurality of different resolution versions, wherein each chunklist of the first respective chunklists includes a first plurality of file segment pointers to first respective file segments for each different resolution version of the first plurality of different resolution versions;**

xiii. **the first respective file segments for each different resolution version of the first plurality of different resolution versions"**

290. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches these elements.

291. In my opinion, the above elements (17.5) refer to nothing more than claiming the contents of a master playlist configured in accordance with the known conventional HLS streaming protocol. To support the HLS format, the master playlists, media/video playlists (chunklists), and corresponding chunks ***must*** conform to the HLS Specification, otherwise the protocol is not HLS. Ex.1052, 4 (§4) ("Playlists that violate these rules are invalid; clients MUST fail to parse them.")

292. For example, Kellicker describes the use of "a master rendition list" or "manifest" (*i.e.*, a portion of 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"). 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 has a rendition chunklist and chunks (segments or fragments). *Id*., ¶¶4, 27. Kellicker further teaches that this protocol applies to both live and VOD streams. *Id*.

(A) Playlist specifying different resolution versions

293. Kellicker further teaches a plurality of renditions that are a plurality of different versions of the same media stream, *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.").

294. Similarly, Pantos recommends including resolution indications in master playlists for streaming video. Ex.1052, 15-17 (§4.3.4.2.) In my opinion, consistent with the HLS specification, it would have been obvious to a POSA to have included resolution variants in the HLS file suite in order 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.

(B) Chunklist pointers to respective chunklists

295. As seen from the exemplary master playlist taught by Pantos and reproduced below, a master HLS playlist includes a corresponding first plurality of chunklist pointers to first respective chunklists. In the example master playlist, 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").

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

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, 28 (§ 8.4).

(C) File segment pointers to respective file segments

297. As discussed 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., 28 (§ 8.2).

298. 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). *See also*, *Id*., 3 (§2) ("A Media Playlist contains a list of Media Segments, which when played sequentially will play the multimedia presentation."), 3 (§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.").

299. For other renditions/resolutions specified in the master playlist (for each different version of the first plurality of different versions), there would be a corresponding set of file segment pointers and file segments in the corresponding media playlist. Ex.1052, 5 (§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").

300. As discussed above with respect to claims 10 and 11 §§X.A.2 and X.A.3.a, 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 server at the peak of the distribution hierarchy of an HLS CDN. The master playlist (master rendition list or portion of an 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 stream. In my opinion, for this reason alone, transfer of these constituent parts of the HLS file suite by RTMP media server and the at least one WebRTC media server to the HLS mother server would have been obvious to a POSA.

301. In addition, configuring the HLS file suite using a master rendition list or master playlist in the manner specified by the HLS Specification 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 consistent with both Kellicker and the HLS Specification.

302. As discussed above, 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 server at the peak of the distribution hierarchy of an HLS CDN. The master playlist (master rendition list or portion of an 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 stream. In my opinion, for this reason alone, transfer of these constituent parts of the HLS file suite by RTMP media server and the at least one WebRTC media server to the HLS mother server would have been obvious to a POSA.

xiv. **Element 17.6: "the HLS mother server caches the first HLS playlist, the first respective chunklists, and the first respective file segments and passes on to at least one HLS child server of the plurality of HLS child servers at least one first copy of the first HLS playlist, the first respective chunklists, and the first respective file segments;"**

303. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches this element.

304. A POSA would have understood that these elements are simply part of the CDN distribution process and architecture taught by Phillips. See §X.A.3.a. *See also* 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).”). This is further confirmed by Ex. 1054 Basile. For example, ¶ 16 of Basile discusses the purpose of the CDN, which is to avoid burdening the origin server with requests, by caching the content:

> In operation, end user device 120-121 request network content, such as content 135 associated with origin server 130. Instead of these requests being handled by the origin server 140, content node 110 of a content delivery network receives the content requests over ones of links 140-141 and processes the content requests for delivery of the content to the associated end user devices 120-121. Requested network content that is already stored in content node 110 can be provided quickly to the end user devices, while network content that is not already stored in content node 110 can be responsively requested by an associated content node 110 from origin server 130 for caching by content node 110. In this manner, content node 110 can act as an intermediary proxy node to provide local and fast access for end user devices 120-121 to network content of origin server 130 without burdening origin server 130 with many end user content requests.
> Ex. 1054 ¶ 16.

305. Consistent with this, Phillips discloses a national server 102 of its CDN network that is an exemplary mother server. Ex.1050, ¶¶31, 32, 56, Fig. 12. In my opinion, a POSA would have understood that the national/mother server of Phillips can act as the primary server where the original/complete version of the served content can be stored, acting as a source for content distributed by the Content Delivery Network (CDN) which can then distribute locally cached copies of that

content from geographically distributed edge servers closer to users for faster access. In other words, a POSA would have understood that the national/mother server of Phillips can act as a backend "mother" server that holds the master copies of content that is served through the CDN network including child servers such as regional servers. Ex.1050, ¶61 ("If the missing audio, video, and/or subtitle asset segments are within the historical delivery trends/thresholds, such segments may be pulled from a regional or national server node or another edge server node (block 1072).").

xv. **Element 17.7: "the at least one HLS child server of the plurality of HLS child servers server caches the at least one first copy of the first HLS playlist, the first respective chunklists, and the first respective file segments and passes on to the HLS CDN at least one second copy of the first HLS playlist, the first respective chunklists, and the first respective file segments; and"**

306. In my opinion, this element is satisfied for the reasons elements 11.0-11.4, 15.0, 17.2 and 17.6 are satisfied. See §§X.A.3.a, X.A.3.b.i., X.A.5.a.v.-viii., xiv. above. Ex.1050, ¶¶8-9, 31, 59.

xvi. **Element 17.8: "the first copy of the first plurality of copies of the first broadcaster's live stream of digital content to the first viewer client device of the first plurality of viewer devices is based on the at least one second copy of the first HLS playlist, the first respective chunklists, and the first respective file segments."**

307. In my opinion, this element is satisfied for the reasons elements 11.0-11.4, 15.0, 17.2 and 17.6 are satisfied. See §§X.A.3.a, X.A.3.b.i., X.A.5.a.v.-viii., xiv. above. Ex.1050, ¶¶8-9, 31, 59.

b. Dependent Claim 18

i. **Elements 18.0-18.1: "the at least one of the at least one RTMP media server and the at least one WebRTC media server is configurable according to a plurality of HLS settings for generating the first HLS file suite, wherein the plurality of HLS settings include: a video segment duration for each file segment of the respective file segments; and"**

308. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches these elements.

309. Pantos teaches that segments contained in HLS playlists have a specified target duration. Ex.1052, 7 (§4.3.2.1) ("The EXTINF tag specifies the duration of a Media Segment. . . . This tag is REQUIRED for each Media Segment."). In my opinion, a POSA would have understood media servers to be "configurable" to set and include a video segment duration in the HLS file suite in order to comply with the HLS protocol. *See also* Ex.1056 Yang (U.S. Patent No.

9,973,785), 20:3-19 ("A segment size is a duration (e.g., length of time) for each MMSF, which may be determined by the ingest manager in order to divide the publishing stream into smaller MMSFs of the same duration. A playlist for a live event may have a specified or target segment size, which may be a specific duration (e.g., 2 seconds) for each MMSF. A maximum segment size may be determined for each live event, where the maximum segment size is the maximum valid segment duration for the live event.").

ii. **Element 18.2: "an HLS buffer length including a first number of the first plurality of file segment pointers in each chunklist;"**

310. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches this element.

311. According to the HLS Specification, as taught in Pantos, the number of file segment pointers in a chunklist contained a media playlist may be limited. Ex.1052, 22 (§ 6.2.2) ("the server MAY limit the availability of Media Segments by removing Media Segments from the Playlist file (§ 6.2.1). If Media Segments are to be removed, the Playlist file MUST contain an EXT--MEDIA-SEQUENCE tag."). In my opinion, a POSA would have understood media servers to 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. *See also* Ex.1056, 19:56-20:2 ("A playlist

length is a number of MMSFs that may be identified, by the media URI for each segment, in the playlist. A target playlist length may be a chosen number of media URIs to add to the playlist for that specific live event. The target playlist length may act as a minimum number of media URIs, such that a playlist may be generated with more than the target number of media URIs, but will not generate a playlist with fewer media URIs than the target playlist length.")

iii. **Element 18.3: "the video segment duration for each file segment of the respective file segments is configured to be approximately two seconds;"**

312. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches this element.

313. Pantos discloses, by way of example, a video segment duration of "less than 10 seconds" or "3.003 seconds." Ex.1052, 3 (§2). Two seconds is both less than 10 seconds and "approximately" 3.003 seconds. In my opinion, 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. *See also* Ex.1056, 3:41-54 ("each playlist contains a set number of MMSF URIs, where each MMSF URI provides the location of the MMSF in a data storage service. The MMSFs may each be of different or equal lengths. In some examples, MMSFs will be between 2-10 seconds in duration."), 20:3-19 ("A playlist for a live event may

have a specified or target segment size, which may be a specific duration (e.g., 2 seconds) for each MMSF.").

iv. **Element 18.4: "the first number of the first plurality of file segment pointers in each chunklist for the HLS buffer length is one of four, five and six."**

314. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches this element.

315. According to the HLS specification, as taught in Pantos, "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, 22 (§6.2.2). Based on the HLS specification, 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 and just above the minimum requirements. In my opinion, 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. ¶. *See also* Ex.1056, 15:36-48 ("For example, the playlist may contain URIs of the four most-recent MMSFs of the live event stored at the data storage service"), 19:56-20:2 ("a target playlist length of four may correspond to four media URIs in the playlist.

Each time the playlist is updated, a number n of the media URIs may be removed from the playlist and a new number z of media URIs may be added to the playlist, where the number of new media URIs added to the playlist must be four or more.").

6. '697 Ground 6: Claims 23 and 24 Are Obvious Over Ellis in View of Spivey

316. For the reasons stated in §IX.A, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker and Phillips to arrive at the claimed invention with a reasonable expectation of success.

a. Independent Claim 23

i. **Element 23.0: "A method for providing a first copy of a first broadcaster's live stream of digital content including first live sporting event video-based commentary from a first broadcaster client device to a first viewer client device, the method comprising:"**

317. Essentially, Claim 23 is a method claim partially implementing the system of Claim 1. Claim 23 generally is broader requiring only a first instance rather than first and second instances (e.g., a first plurality of copies of a first broadcaster's live stream of digital content including first live sporting event video-based commentary from a first broadcaster client device to a first plurality of viewer client devices). *See* Ex.1006, Claims 1, 23. Regardless, Ellis in view of Spivey discloses a system where there are multiple live streams of digital content about multiple different live sporting events. Thus, a POSA would have understood that

Ellis in view of Spivey discloses this element for the same reasons stated above in connection with claim element 1.0. §X.A.1.a.i., *supra*.

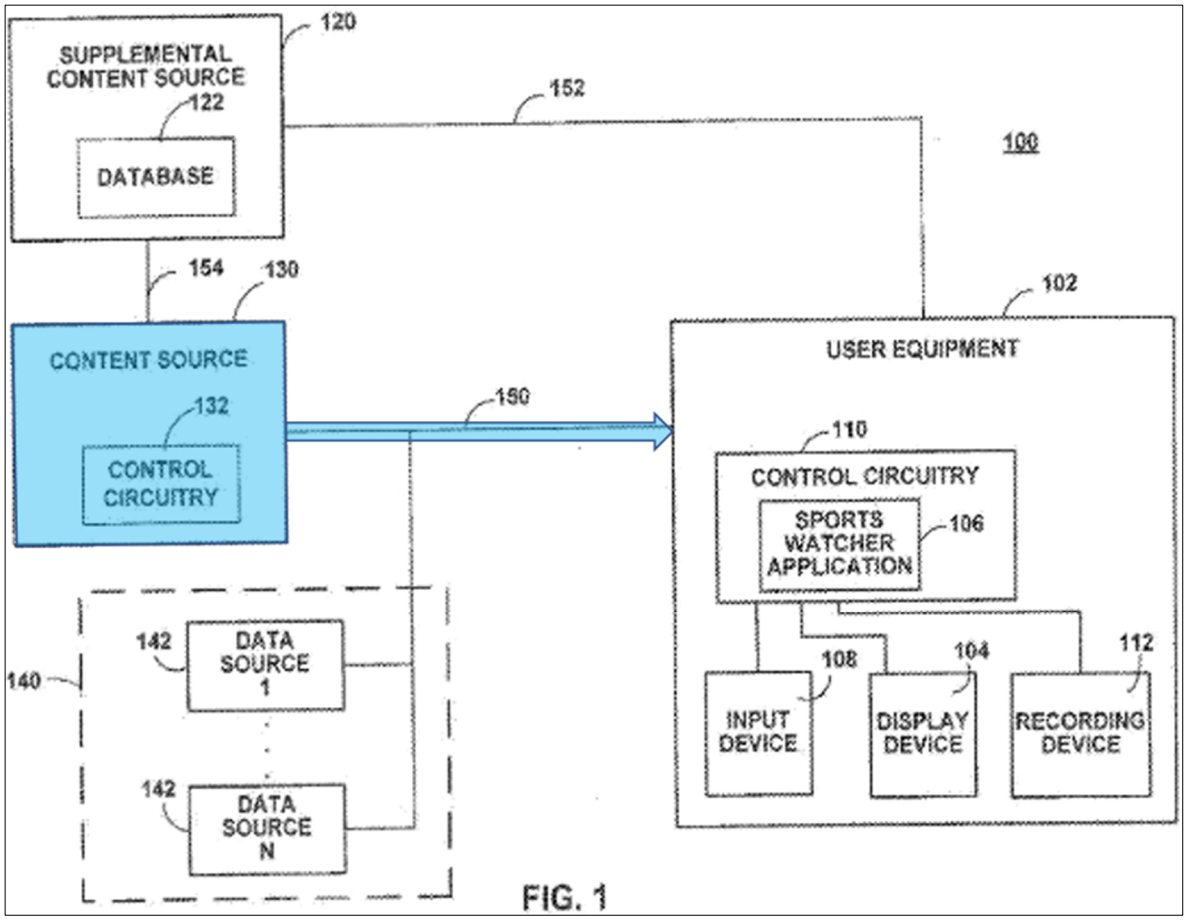
318. To the extent the preamble is limiting, it is my opinion that Ellis discloses this element. *See* Ex.1016, Abstract. Ellis discloses providing a first copy of a first broadcaster's live stream of digital content including first live sporting event video-based commentary from a first broadcaster client device to a first VCD by providing "enhanced information access and viewing functionality for sports-related programming," which includes "a sporting event, sports news, or sports commentary." Ex.1016 ¶¶8, 11, 38-40.

319. Ellis discloses providing the "media content" from those sporting events "to user equipment 102." *Id.* ¶¶38–40; Fig.1. Ellis defines "sporting event" as encompassing "online sports commentary." *Id.* ¶38. And, Ellis teaches providing sports commentary in a data feed or over communications channel to a user. *Id*., ¶41. Further, media content includes broadcast programming "and any other content capable of being displayed by, presented to, … or interacted with, using user equipment 102." *Id.* ¶40. A POSA would have readily appreciated that broadcasts of live events would obviously, if not implicitly, also include video-based commentary. Ex.1016, ¶38. Additionally, "commentary" is natively supported in HLS live streams as alternative audio to an associated video stream (i.e., video-based commentary). Ex.1052, 28-29 (§8.6).

ii. **Element 23.1: "A) providing the first copy to the first viewer client device via a first video communication channel between a plurality of media sources and the first viewer client device;"**

320. In my opinion, Ellis and Spivey disclose this element. *See* Elements §§X.A.1.a.i.-iv. *supra*. These elements are substantially the same as the '697's element 1.3. Ex.1003, Claims 1, 23. The claim language is slightly altered, but the claims contain the same elements. Thus, a POSA would have understood that Ellis and Spivey disclose this element for the same reasons stated above in connection with at least claim elements 1.0-1.3.

321. Ellis discloses content sources, which are "any suitable source of media content" such as an "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 broadcaster's live stream of digital content and provide video content to user equipment via communications paths. *See id.* ¶¶39-41, 45, 47. Figure 1 shows a content source providing video content to user equipment over communication path 150:



Ex.1016 ¶¶39-41, Fig. 1 (annotated). Ellis also discloses multiple content sources, *i.e.*, a plurality. *Id.* ¶39. ("[T]here may be numerous instances of content source 130 …"); ¶74. As also apparent from Fig. 1 above, Ellis also discloses that users can be connected with a plurality of data sources 140. *Id*. Moreover, as explained below, Ellis's content sources perform the same functions as the media sources as recited in 23.1.

322. As explained above, Ellis discloses providing individual copies to live streams to client devices. Ellis further discloses providing digital content or video content over a video communication channel. As an example of an "video

communication channel," '697 describes a "video" channel between a media server and a client device to convey video-based commentary. Ex.1006, 5:36-49.

323. Ellis discloses providing video from a content source to user equipment, over a communication path. Ex.1016 ¶39, 74. The content source can be an "internet or network media/web server" and the user equipment "may include computer equipment." *Id.* ¶¶40, 48. The communication path can be "for example, a satellite path, a fiber-optic path, a cable path, or any other suitable wired or wireless path." *Id.* ¶45. 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 is provided with a copy from the content source over a first video communication channel that is unique to each user device. *Id.*

iii. **Element 23.2: "B) periodically retrieving, via the Internet and from an event information provider, first event information germane to a first live sporting event, wherein the first event information includes first score information for the first live sporting event; and"**

324. In my opinion, Ellis and Spivey both disclose this element. This element is substantially the same as the '697's element 1.5. Ellis and Spivey both disclose a multiple-sporting-event system that retrieves real-time data for all events.

§X.A.1.a.v. *supra*. The claim language is slightly altered, but the claims contain the same elements. Thus, a POSA would have understood that Ellis in view of Spivey discloses this element for the same reasons stated above. §§X.A1.a.v. *supra.*

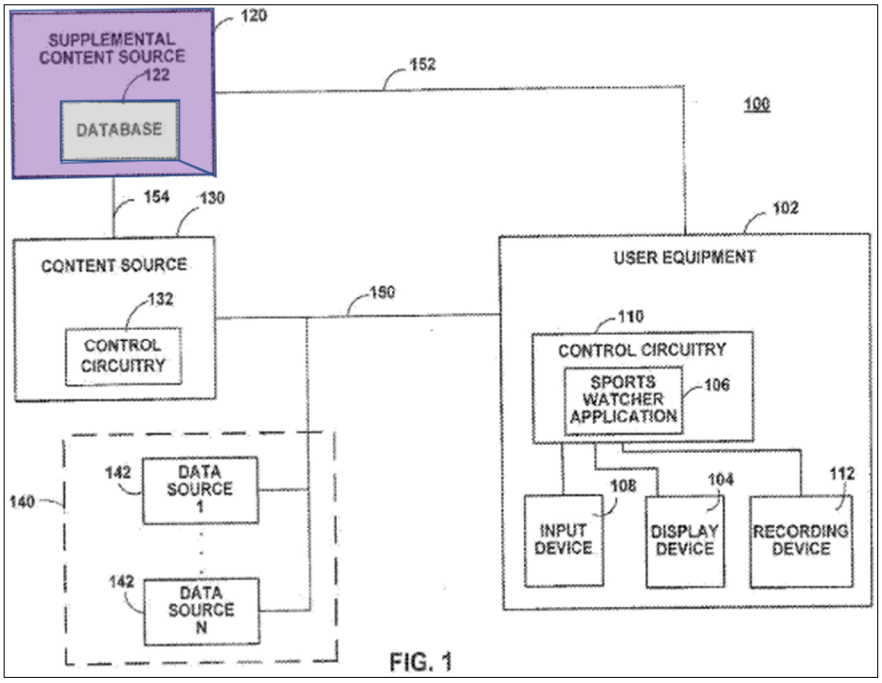
325. The '697 explains that the control server is a generic server comprising one or more processors, memory devices, and communication interfaces. Ex.1006, 55:60-56:3. The control server retrieves "event information" from an "event information provider." *Id.*, 21:37-42. The control server then transmits the event information to the socket server. *Id.*, 19:41-45, 21:37-48. 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.*, 19:27-37, 21:37-42. Further, the control server may "store at least some portion of retrieved event information." *Id.*, 45:20-48.

326. Ellis's SCS comprises one or more processors, memory devices, and communication interfaces. 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." Ex.1016 ¶¶42, 142. The SCS periodically retrieves, via the internet and from an event information provider, first event information germane to multiple live sporting events, including score information for each of the multiple live sporting events because the SCS may "provide real-time sports scores and game summary information, … spread and wagering data, and any other sports-related information

to user equipment 102." *Id.* ¶41; see *id.* ¶¶72, 117. In my opinion, a POSA would have understood that this information is retrieved via the internet. *Id*. ¶¶37-38, 40, 142. Furthermore, a POSA would have understood that this was how real-time data was routinely transferred to users at the time of Ellis and Spivey—including score updates and wagering information. Ex.1016 ¶¶42, 142. Additionally, in my opinion, 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. Ex. 1024, 408-410.

327. While a periodic time interval, *e.g.*, every five seconds, is not explicitly mentioned in Ellis, it is my opinion that 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. 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, yet to limit queries to the system and load on the components. In my opinion, a POSA would have recognized the period of retrieval should be tailored to the nature of the event and/or the use case. For example, polling for scoring updates on a basketball game at a frequency of 1000 times a second may provide little benefit where the frequency of scoring is tied closely to a 24-second shot clock.

328. 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).

329. To the extent Ellis does not expressly disclose or render obvious this element, it is my opinion that Spivey's live data server device ("LDSD"), as shown in Figure 4, contains the same basic components as the '697's control server. Ex.1017, Fig. 4.

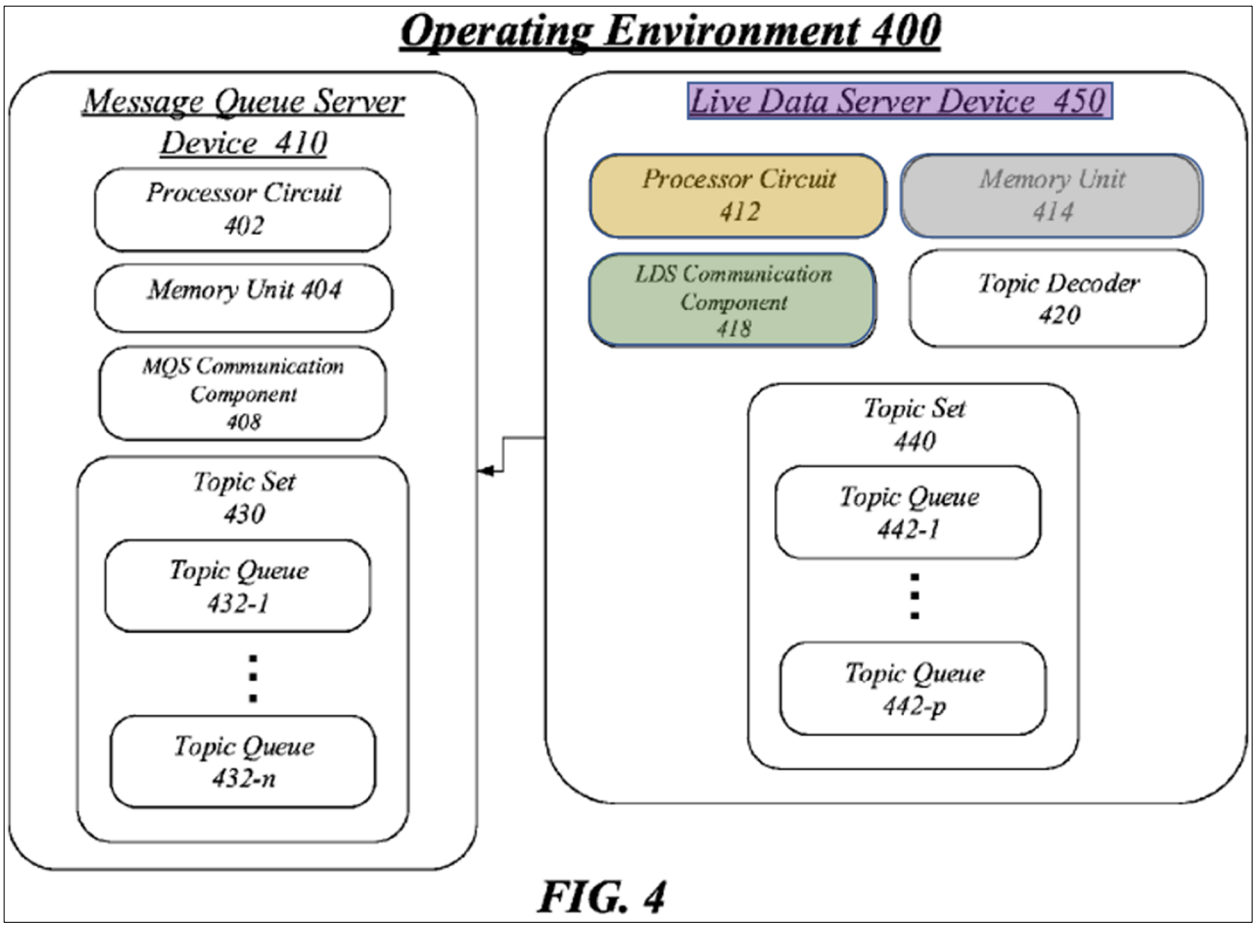


FIG. 4

Ex.1017, ¶¶54-55, Fig. 4 (annotated).

330. The LDSD is a generic server comprising one or more processors, memory devices, and communication interfaces. *Id.* In my opinion, the LDSD performs the same functions as the '697's control server. The LDSD contains an LDS communication component that allows LDSD "to receive live data and transmit the data payloads to MQS device 410." *Id.* ¶55. The live data received by the LDSD includes first event information related to at least first live event, e.g., "sports scores and breaking news" and when related to baseball can include "a score, a play, an error, an injury, a player-at-bat, and so forth." *See id.* ¶¶27, 48, 66 ("A data payload

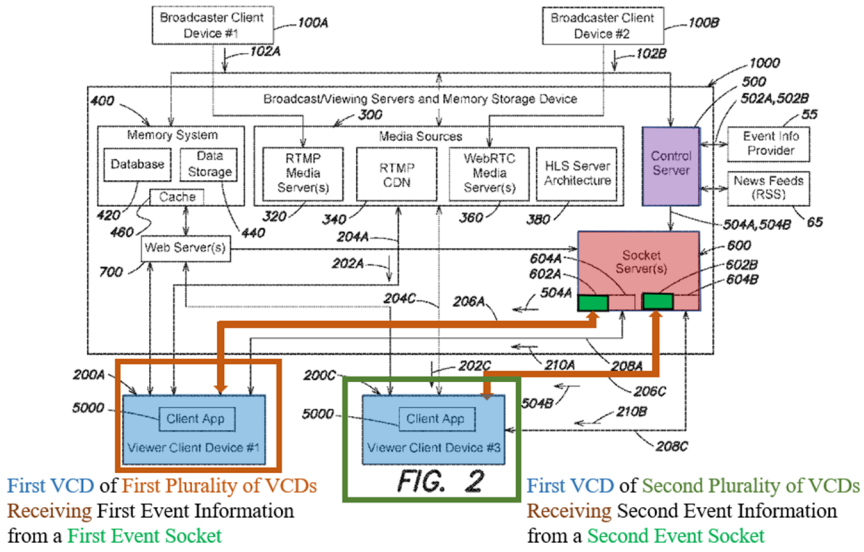
may include … information about the live event in progress, such as a score, a name of the player that scored or made a play, an injury, a penalty, an error, a new record set, and so forth."), Fig. 9. The LDSD retrieves the first event information germane to a first live sporting event including the 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. *Id*. ¶¶35, 37, 178. Thus, it is my opinion that a POSA would have understood that Ellis and Spivey both disclose this element. To the extent Ellis's SCS did not meet this claim element, it is my opinion that a POSA would have replaced Ellis's SCS with Spivey's LDSD and MQSD to decrease latency. *See* §§IX.A, X.A.1.v. *supra*.

iv. **Element 23.3: "C) transmitting the first score information to at least the first viewer client device via a first event information communication channel between at least one first event socket of at least one socket server and the first viewer client device of the first plurality of viewer devices, wherein the at least one first event socket corresponds to the first event information germane to the first live sporting event."**

331. In my opinion, Ellis in view of Spivey discloses these elements. These elements are substantially the same as the '697's elements 1.8 except that there is a different name for the communication channel. Ex.1006, Claims 1, 23. The different channel name alters nothing. §X.A.1.vi-viii *supra.* Ellis in view of Spivey

discloses a system of providing real-time data from multiple live games to multiple groups of users over separate (third/fourth) internet connection feeds. §§§X.A.1.vi-viii. The claim language is slightly altered, but the claims contain the same elements. Thus, a POSA would have understood that Ellis in view of Spivey discloses these elements for the same reasons as stated above. §§§X.A.1.vi-viii, §X.A.1.vi-viii, *supra*.

332. This element requires transmitting score information about the first live event from the first socket to the VCD. *See* Ex.1006, Claim 23. These features are shown in Figure 2:



Id., Fig. 2.

333. In my opinion, Ellis in view of Spivey discloses a socket server communicatively coupled to the control server (which, as discussed above, is disclosed by Ellis's SCS and Spivey's LDSD). The '697 describes the socket server

as a generic server. *See* Ex.1006, 55:60-6:3. The purpose of the socket server is to "facilitate communication" of "synchronized event information retrieved by the control server" associated with an event between the control server and the VCDs. *Id.*, 19:64-20:5. 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 the event information in "event messages" via the event information channel "in a synchronized manner" to all viewers "following the particular event." *Id.*

334. Ellis discloses providing event information over a separate connection to user equipment. Ex.1016 ¶¶39-41, Fig. 1. Ellis also discloses a system where multiple servers provide the event information. Ex.1016 ¶47 (noting the SCS's functionality "may be … provided by multiple systems at multiple facilities").

335. In my opinion, a POSA would have looked to Spivey to add additional backend architecture to reduce latency in the real-time data delivery system. Like the '697'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, it is my opinion that a POSA would have understood that Spivey's MQSD carries out the same purpose as the '697's socket server. The MQSD "receive[s]" and "process[es] live data to provide live data" to "the subscribed

connected client devices substantially simultaneously." *Id.* ¶¶36; *see id.* ¶¶45 ("[MQS] device 310 may be a representative example of message queue server 120"), ¶54 (describing that 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:"

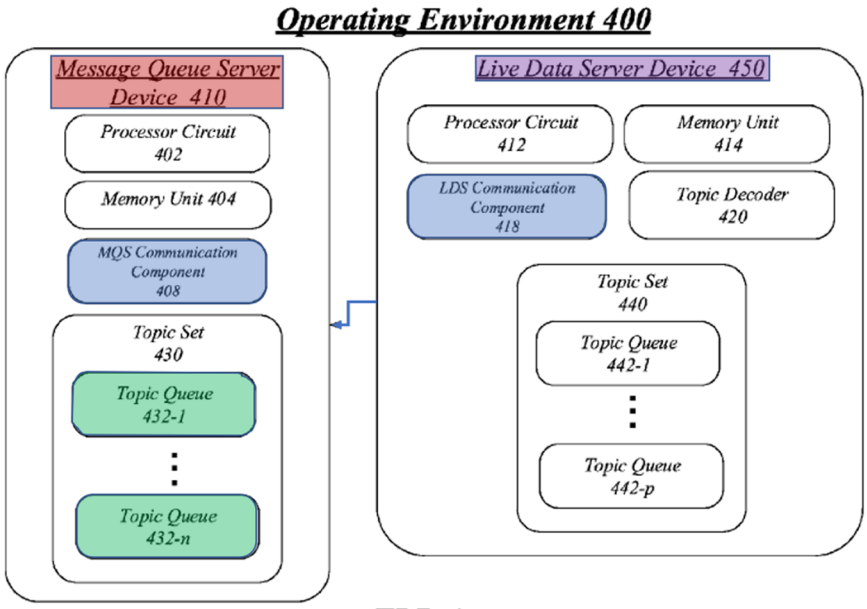


FIG. 4

Ex.1017, ¶¶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 (disclosing that communication component 308 transmits and receives data), ¶54 (disclosing that communication component 408 is analogous to 308), ¶55 ("LDS communication component 418 … allows [LDSD] to … transmit the data payloads to [MQSD]."). Then, 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. After sorting the data into topic queues, the MQSD shares the information "substantially simultaneously" to "the subscribed connected client devices." *See id.* ¶96. Thus, it is my opinion that a POSA would have understood Spivey's MQSD discloses the '697's socket server being communicatively coupled to a control server.

336. Alternatively, Spivey teaches that latency associated with the broadcast of event data, such as sports scores, to clients can be reduced by "using topic cues and open duplex communication connections." *Id.* ¶¶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."). As an example of such "open duplex communication connections" Spivey discloses that communications between a server and a client 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"); *see also, Id.*¶¶ 43, 61, 74, and 92; claim 8. Thus, in my opinion, it would also have been obvious to a POSA in view of these teachings of Spivey 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. In my opinion, a POSA would have

recognized that doing so would further the express goals of Spivey with respect to reduction of latency in such transmissions. In my opinion, a POSA would have had a reasonable expectation of success implementing this web socket functionality in the MQSD because this would have involved nothing more than a combination of well-known, compatible elements to achieve a predictable result.

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

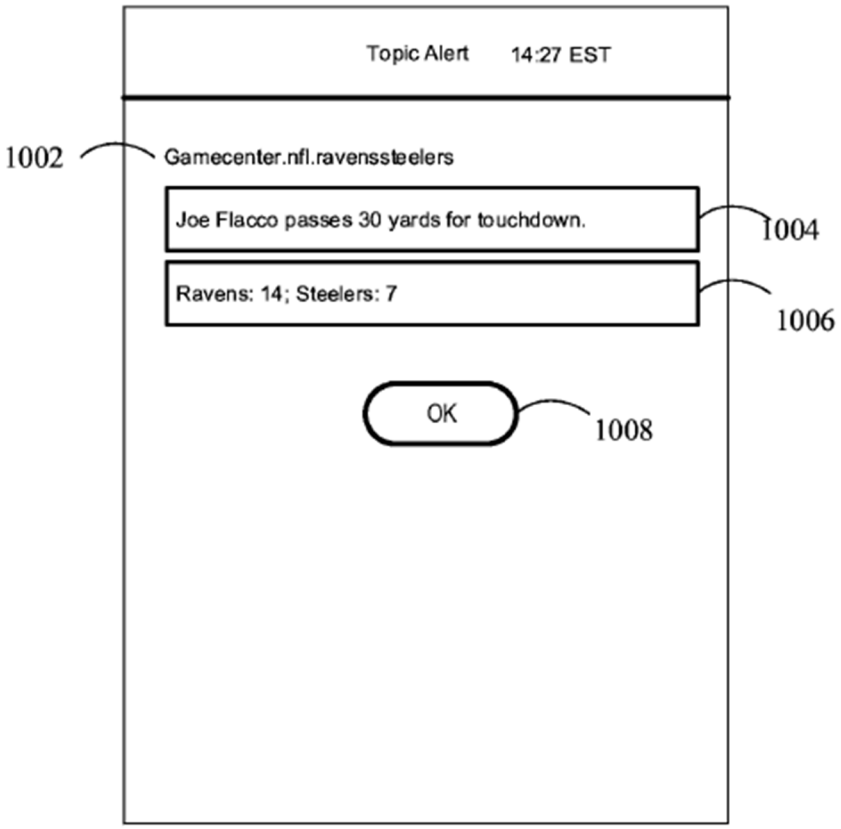


FIG. 10

Id. ¶85, 117, Figs. 9-10. Thus, it is my opinion that a POSA would have understood that Spivey discloses a socket server receiving from the control server at least the first score information.

(A) First event socket corresponds to the first event information germane to the first live sporting event

338. Spivey often discusses a connection between the client device and the system using a live data access server ("LDAS"), however, Spivey also discloses and teaches that the message queue server device may be "accessible to clients over the network directly." Ex.1017 ¶¶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. Also, the system does not require an LDAS, only that it "may include a[n] [LDAS]." *Id.* ¶38. Thus, it is my opinion that a POSA would have understood Spivey to disclose and teach a direct connection between the client devices and MQSD. In my opinion, a POSA would have used this implementation because reducing the amount of server transfers would reduce latency, which is Spivey's main purpose. Furthermore, it was a well-known alternative to a POSA to implement the functionality of multiple servers as a single server. Ex.1016, ¶¶4-7.

339. In my opinion, a POSA would have understood that 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. *Id.* It is my opinion that 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. *See* Ex.1006 at 20:45-61, 21:53-22:6); Ex.1029, pp. 14-20; Ex1030; §IV.C.3., *supra*.

340. In the same way, 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 then 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.

341. 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. *See 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:

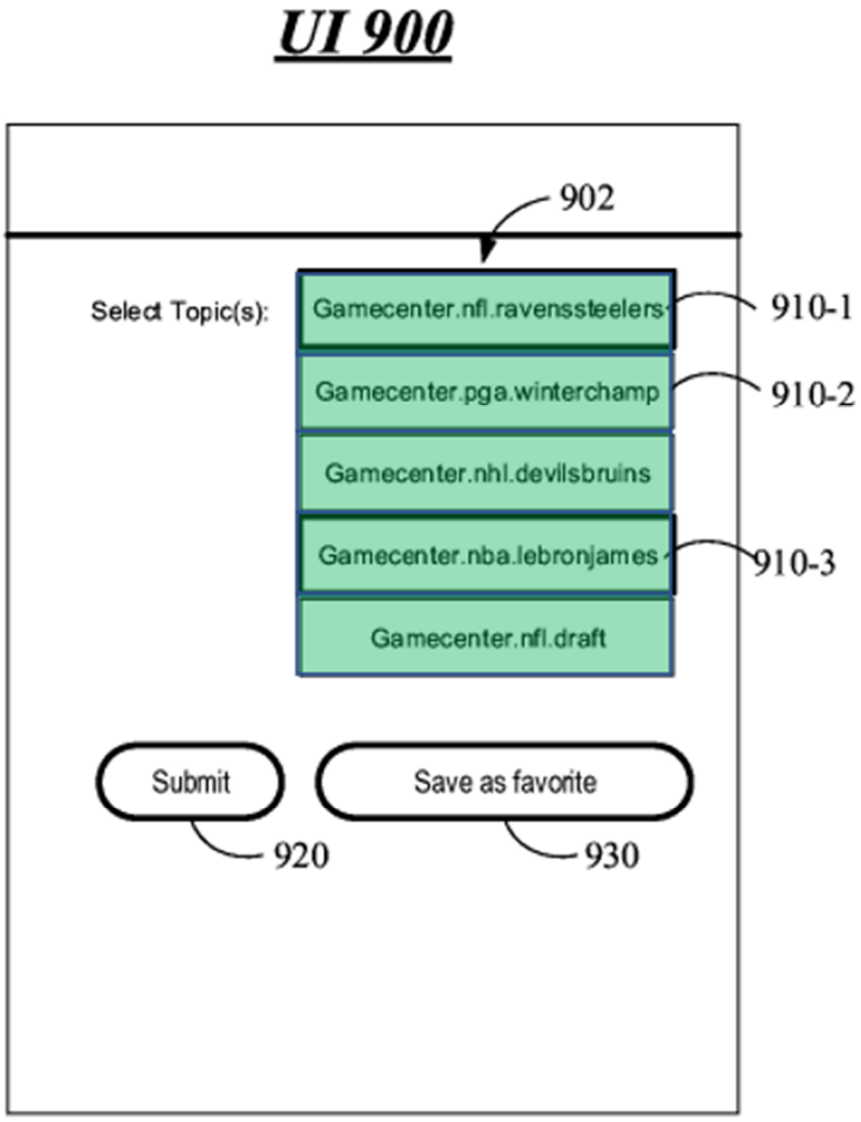


FIG. 9

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, it is my opinion that a POSA would have understood that Spivey's topic queues disclose the

'697's first event sockets corresponding to the first event information germane to the first live sporting event.

342. Alternatively, it would also have been obvious to a POSA in view of these teachings of Spivey 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 in §IX.A. and §X.A.1.a.vi.-viii.

(B) Transmitting the first score information to at least the first VCD

343. In my opinion, Ellis discloses a first VCD. *See* §§X.A.6.a.i-ii, X.A.1a.iv. *supra.* Both Ellis and Spivey further disclose transmitting the first score information to the first 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. In Ellis, SCS may be configured "to provide real-time sports scores and game summary information, … to user equipment 102." Ex.1016 ¶41. Thus, it is my opinion that Ellis and Spivey disclose the above element.

(C) Transmitting via a first event information 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

344. As previously noted, Ellis discloses at least a first VCD. *See* §§X.A.6.a.i-ii, X.A.1a.iv., *supra.* However, to the extent that the claim is interpreted as reciting "***a*** *first plurality of viewer client devices*," Ellis discloses multiple broadcasters providing copies of each broadcaster's live stream of digital content including live sporting event video-based commentary 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. 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. *Id.* ¶45, 74. The communication path can be any "suitable wired or wireless path." *Id.* "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 from a plurality of broadcaster devices.

345. In my opinion, Ellis in view of Spivey discloses providing first event/score information over a separate 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 Internet." *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 than the video signal 150. Ex.1017 ¶39-41, Fig. 1. Thus, in my opinion a POSA would have understood Ellis in view of Spivey to disclose transmitting first event information via a separate internet connection, which includes the first event information communication channel.

346. Further, this information is provided via a connection between a topic queue and the first user equipment device of the first plurality of user devices. Users can access the MQSD using a URL. *See* Ex.1017 ¶¶81-82. Then, the user can access the specific topic queue she wishes to receive information from by using a text string, *e.g.*, a topic label or Event ID, associated with a topic queue. *Id.* ¶179, Fig. 9

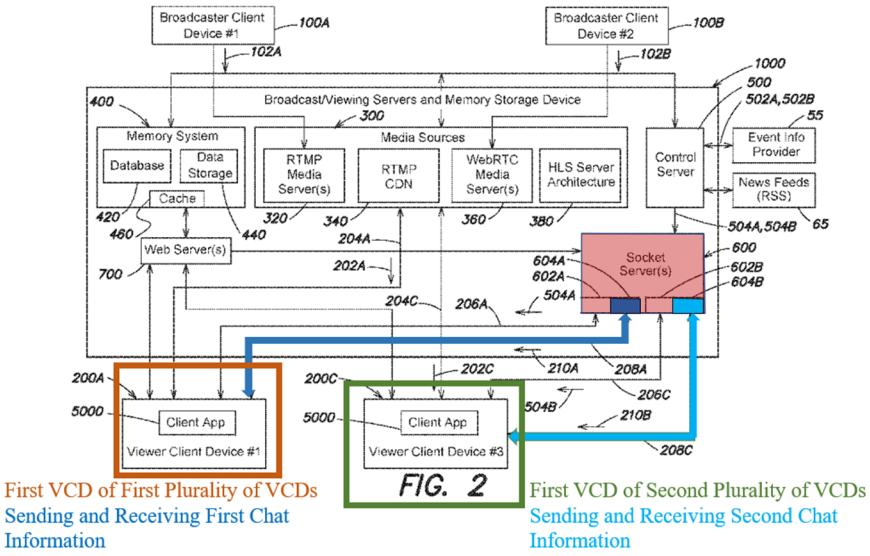
("Gamecenter.nfl.ravenssteelers."). Thus, in my opinion a POSA would have understood that Ellis in view of Spivey discloses distributing event information from a first event socket of the socket server to a first user device of a first plurality of user devices over an internet connection. Further, it is my opinion that 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.

b. Dependent Claim 24

i. **Elements 24.0-24.1: "The method of claim 23, further comprising: D) transmitting and receiving first chat information regarding the first live sporting event video-based commentary included in the first broadcaster's live stream of digital content via at least one first chat/system event Internet communication channel between at least one first chat/system event socket of the at least one socket server and the first viewer client device of the first plurality of viewer client devices, wherein the at least one first chat/system event socket corresponds to the first chat information regarding the first live sporting event video-based commentary."**

347. In my opinion, Ellis in view of Spivey discloses these elements. Ellis in view of Spivey discloses transmitting and receiving chat information regarding the first live sporting event video-based commentary via individual communication channels between a first socket and a first VCD of a first plurality of VCDs. §§X.A.4.c, *supra*.

348. The '697 explains that a chat/system event communication channel may be any generic internet communication channel. Ex.1006, 5:29-36. These elements only further require that the system "transmits and receives" first "chat information regarding" the first "live sporting event video-based commentary included in the first stream of digital content" over another separate, individual connection between a first "chat/system event socket" and a first VCD:



Ex.1006, 22:7-39, Claim 24, Fig. 2. In sum, it is my opinion that a POSA would have understood that these elements require that chat information is transmitted and received over a communication channel from another socket that is dedicated for chat/system information.

349. Ellis discloses sending and receiving chat information because users "may also leave feedback regarding the commentary that other users in the media system may view when they access the same commentary." Ex.1016 ¶132. Such

commentary includes "text." *Id*., ¶132. Thus, it is my opinion that a POSA would have understood that Ellis discloses a system that transmits and receives first chat information regarding the first/ live sporting event video-based commentary included in the first broadcaster's live stream of digital content. *See also* Ex.1016 ¶126 ("To create and send a message (e.g., instant message or email message), the user may select send message option.").

350. Ellis also discloses providing this information on a separate channel to users. 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. *Id*. Such commentary includes "text." *Id*., ¶132. Thus, it is my opinion that a POSA would have understood that Ellis discloses transmitting via at least one first chat/system event Internet communication channel between at least one first chat/system event socket of the at least one socket server and the first VCD. To the extent that the claim could also be interpreted as reciting "***a*** *first plurality of viewer client devices*," Ellis discloses this aspect as explained at §X.A.6.a.iv.(C).

351. Ellis in view of Spivey discloses a socket specifically designated for chat/system information. Spivey utilizes topic queues to organize "data payloads"

that "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 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. 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." In sum, the statistics would have been separate from the chat information. Thus, a POSA would have understood Spivey to teach a first chat/system event socket (*e.g.*, "gamecenter.nfl.ravenssteelers.chat") corresponding to the first chat information regarding the first live sporting event video-based commentary.

7. '697 Ground 7: Claim 26 is Obvious Over Ellis in View of Spivey, Kellicker, and Phillips

352. For the reasons stated in §IX.C, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker and Phillips to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 26

i. **Elements 26.0-26.2: "The method of claim 24, wherein the plurality of media sources comprises: at least one of: at least one real-time messaging protocol (RTMP) media server to receive the first broadcaster's live stream of digital content; and at least one web real time communication (WebRTC) media server to receive the first broadcaster's live stream of digital content; and"**

353. In my opinion, Ellis in view of Spivey and Kellicker teaches these elements. The above elements are essentially the same as claim elements 10.0-10.2. Therefore, for at least the reasons explained above at §X.A.2.a.i. above, these elements would have been obvious to a POSA.

354. As explained in §IX.B above, Kellicker discloses a server (120/1250) that receives incoming live media streams (e.g., live incoming media stream (1220)), processes the incoming media streams and serves such media streams to playback devices in real-time. Ex.1049, ¶¶100-101, 106. Kellicker discloses that this server can be implemented through a plurality of such 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"), ¶24. Thus, Kellicker discloses a plurality of media sources in the form of real-time streaming media servers.

355. Kellicker also teaches that the server(s) may support various streaming protocols including, RTMP, and WebRTC. *Id*., ¶106. ("computing device 1250 may also support real time messaging protocol (***RTMP***) (and variants thereof), real-time streaming protocol (RTSP), real-time transport protocol (RTTP), web real-time communication (***WebRTC***), and MPEG-2 transport stream (MPEG-TS)"), ¶ 24 (disclosing transcoders/transmuxers for transforming media streams). Thus, Kellicker discloses media sources in the form of a real-time RTMP media server and/or a real-time WebRTC server.

356. Kellicker teaches providing its server(s) with the capability to "enable the computing device 1250 to process data in accordance with multiple coding technologies and protocols." *Id*., ¶103-104. In my opinion, it would have been obvious to a POSA to have implemented the content source(s) 130 of Ellis in the form of a real-time RTMP media server and/or a real-time WebRTC server. In my opinion, 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.

357. To the extent required by the claims, in my opinion it would have also been obvious to a POSA to have modified Ellis to implement the content source(s) 130 in the form of multiple servers, including a RTMP server ***and*** a WebRTC server. A POSA would have recognized that the use of multiple servers, as taught by

Kellicker, would have provided the system with even more flexibility, capacity and adaptability to handle multiple different types of incoming media and comply with multiple types of protocols. *Id*., ¶98.

ii. **Element 26.3: "an HTTP live streaming (HLS) caching server architecture communicatively coupled to at least one of the at least one RTMP media server and the at least one WebRTC media server,"**

358. In my opinion, Ellis in view of Spivey and Kellicker teaches this element. The above element is essentially the same as claim element 10.3. Therefore, for at least the reasons explained above at §X.A.2.a.ii. above, this element would have been obvious to a POSA.

359. To reiterate, Kellicker teaches that the at least one RTMP media server and/or the at least one WebRTC media server are communicatively coupled to one or more other servers 1280. Ex.1049, ¶102. For example, 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., an edge server), etc. *Id*.; FIG. 12. Kellicker also explains that the data transmitted to the servers 1280 includes "media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof)." *Id*. Thus, Kellicker teaches that

the RTMP media server and/or the WebRTC media server are communicatively coupled to the one or more other servers 1280.

360. Kellicker further discloses that the media streams received by the one or more other servers 1280 "may be associated with the same encoding format and transmission protocol or may be associated with different encoding formats and transmission protocols." *Id*. and ¶24 (disclosing transcoders/transmuxers for transforming media streams). Kellicker also specifically teaches providing the real-time streaming server(s) with the HLS streaming protocol ability. *Id*. at ¶105-106. In my opinion, it would have been obvious to a POSA to implement such HLS conversion capability in the form of the one or more other servers 1280 for Ellis' content source(s). A POSA would have been motivated to provide HTTP live streaming (HLS) media streams because, as was well-known to a POSA, HLS "provides a reliable, cost-effective means of delivering continuous and long-form video over the Internet." Ex.1052, 2 (§1).

361. 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." *Id*. at ¶102. In my opinion, a POSA would have understood a CDN to be a caching architecture. Ex.1052, 2 (§1) ("[HLS] offers compatibility with large-scale HTTP caching infrastructure to support delivery to large audiences"); Ex.1055 Lepeska, 7:19-23 ("[a] content delivery network or content distribution network

(CDN) is traditionally known as a system of computer containing copies of data placed at various nodes of a network, such that access to the cached data is improved by increasing bandwidth and reducing access latency"). To a POSA, the fundamental purpose of a CDN is to provide caching capability in order to improve responsiveness to client device requests for served data (in this case streamed media). *See also,* Ex.1050.

iii. **Element 26.4-26.5: "wherein the HLS caching server architecture comprises: an HLS mother server communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server to receive, from the at least one of the at least one RTMP media server and the at least one WebRTC media server, a first HLS file suite representing the first broadcaster's live stream of digital content; and**

iv. **"a plurality of HLS child servers communicatively coupled to the HLS mother server and an HLS content delivery network (HLS CDN);"**

362. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches this element. The above elements are essentially the same as claim elements 11.1-11.2. Therefore, for at least the reasons explained above at §X.A.3.a.i. above, these elements would have been obvious to a POSA.

363. To reiterate, as explained above at §X.A.2., Kellicker teaches an HTTP live streaming (HLS) caching server architecture including a CDN communicatively

coupled to at least one RTMP media server and/or at least one WebRTC media server. It would have been obvious to a POSA to have provided this HLS caching server CDN 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." Ex.1050, ¶31. Consistent with the teachings of Phillips, a POSA would have been motivated to provide this server architecture in order to provide "for high-performance streaming . . . to subscribers using one or more Internet-based infrastructures." *Id*., ¶30. For example, Phillips teaches that 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 implement Phillips' 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 . . .").

364. 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, coupled to an HLS CDN architecture (1282), and from the CDN to a client/user device (1270):

365.

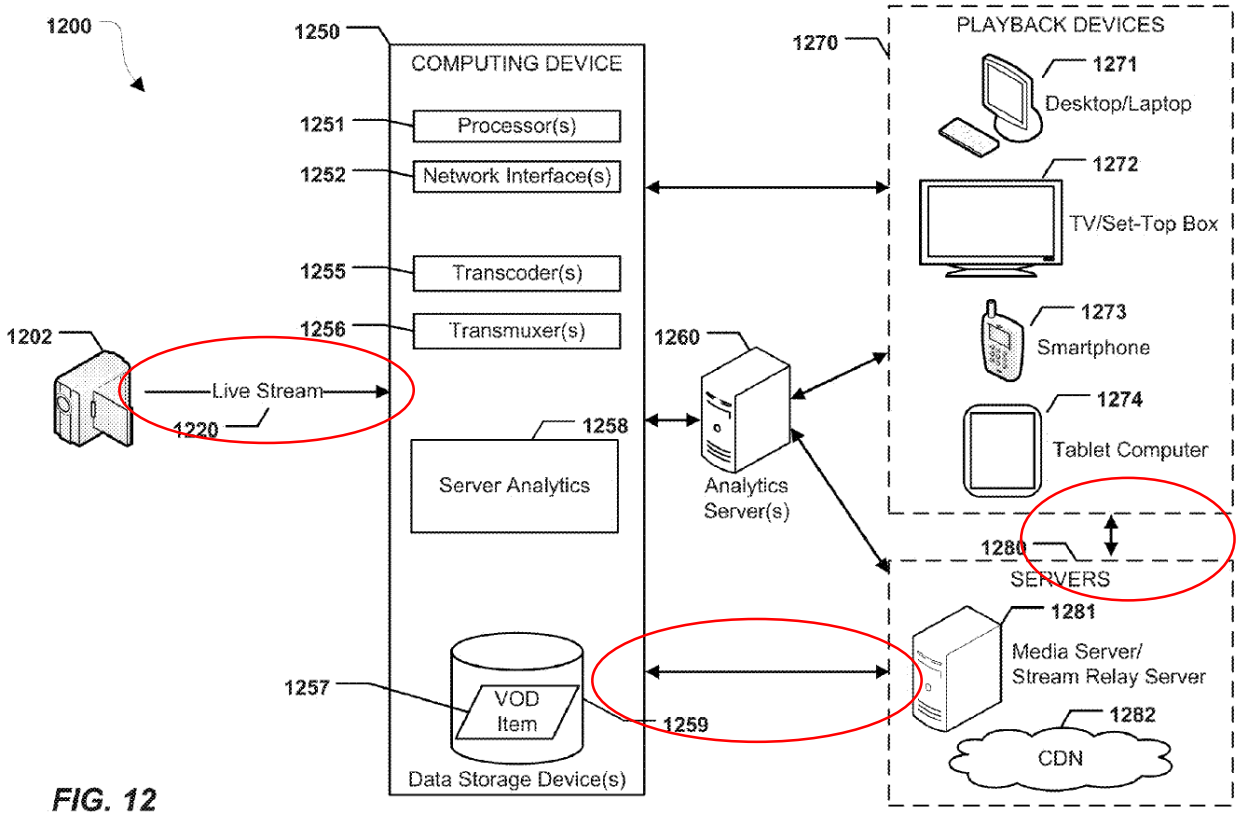


FIG. 12

Ex.1049, Fig.12 (annotated).

366. In particular, Kellicker discloses that the media server(s) 1250 is configured to transmit data to "a server of a content distribution network (CDN) 1282." Id., ¶102. Further, Kellicker discloses that, "[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.

367. In similar fashion, Phillips teaches that content that is 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 that 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 contents and/or nature of the files being handled by the system could be 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.1050, §§8.4-8.6. As can be seen, Phillips' 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.

368. With respect to the "HLS file suite" this represents nothing more than conventional HLS file transfer protocol, as acknowledged in the '697. Ex.1006, 38:47-55. 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.1050, §§8.1-8.7. By delivering content from Kellicker's server 1250 to a Phillips-type CDN, the HLS caching server architecture comprises: an HLS mother server (e.g., national server, regional servers) which are communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server.

369. Hence, 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 POSA would have understood that 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. As explained in the '697, an "HLS file suite" represents nothing more than the contents of a conventional HLS file transfer protocol. Ex.1006, 38:47–55. In my opinion, 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.1052, 27-29 (§ 8.1-8.7).

370. Alternatively, and to the extent that the nature or content of the data could be considered as limiting the claimed system, 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.1050, §§8.1-8.7. For example, as explained in Kellicker, this HLS caching server architecture receives a portion of the first HLS file suite (video stream) representing the first live stream of digital content. Ex.1049, ¶102 ("data transmitted to the playback devices 1270 and to the servers 1280 includes media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof"). Further, Kellicker discloses that "first HTTP port(s) may be used to communicate a master rendition list and media chunks . . .," each of which is a portion of the HLS file suite. Ex.1049, ¶110; *see also* Ex.1006, 38:47–55.

371. Similarly, Phillips teaches that the first HLS file suite is received (media chunks and manifest/playlist). For example, Phillips teaches that content that is requested for delivery by the CDN may include a received "manifest" that is made available to a "content delivery node" (e.g., "mother" or "child" server) of the CDN server architecture. Ex.1050, 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"). And, to the extent that 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

(e.g., a "child" server). Ex.1050, Abstract, ¶31, Fig. 1. As another example, Phillips explains that the media portions of the HLS file suite are received by the CDN architecture. Ex.1050, FIG. 12 ("Pull Missing Segments for ABR Assets", which are media chunks of the HLS file suite), ¶9.

v. **Element 26.6: "A) comprises providing the first copy to the first viewer client device via the first video communication channel between the HLS CDN and the first viewer client device, wherein the first copy is based on the first HLS file suite received by the HLS mother server."**

372. This element is essentially the same as elements 11.3-11.4. Thus, these elements would have been obvious to a POSA for lease the same reasons set forth at §X.A.3.a.ii.-iii above. As explained above in §X.A.6.a.i.-ii, Ellis as modified by Spivey establishes the first video communication channel to provide the first copy of digital content to the first viewer client device. As further explained at §§X.A.2.a.iii-iv, X.A.3.a.ii-iii, Ellis as modified by Spivey, Kellicker and Phillips establishes that the first video Internet communication channels are provided between the first viewer client device and the HLS CDN. *See also* Ex.1050, ¶33 ("Each local edge server is operable to serve a plurality of subscribers in a serving area for delivering content to associated end stations or client devices, as illustrated by an example client device or user equipment (UE) device 105 that may for consume or access streaming content delivered via CDN 100 in any type or number of access technologies including broadband access via wired and/or wireless (radio)

communications."), ¶28 ("client devices may also access or consume content/services provided over… packet-switched wide area public network such as the Internet via suitable service provider access networks.").

373. As explained at §X.A.3.a.i., a POSA would have understood that 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 as required by the HLS specification and the CDN distribution process.

8. '697 Ground 8: Claims 23-24 Are Obvious Over Herzog in View of Spivey and Ellis

374. For the reasons stated in §IX.F, above, it is my opinion that a POSA would have been motivated to combine the teachings of Herzog, Ellis and Spivey, to arrive at the claimed invention with a reasonable expectation of success.

a. Independent Claim 23

i. Element 23.0: "A method for providing a first copy of a first broadcaster's live stream of digital content including first live sporting event video-based commentary from a first broadcaster client device to a first viewer client device, the method comprising:"

375. To the extent the preamble is limiting, it is my opinion that Herzog in view of Ellis discloses this limitation.

376. Herzog provides "live video and audio data" to user devices over a "communication network." *See* Ex.1015, Abstract. Herzog improves on traditional

"sports application systems" where the meta-data is included "as part of the video broadcast." Ex.1015 ¶83, FIG. 4. It is my opinion that a POSA would have recognized that any such system of video broadcasts would obviously, if not inherently, include traditional broadcasts of sporting events that include broadcaster's video-based commentary. Additionally, "commentary" is natively supported in HLS live streams as alternative audio to an associated video stream (i.e., video-based commentary). Ex.1052, 28-29 (§8.6).

377. Similarly, Ellis discloses a system that provides the "media content" from those sporting events "to user equipment 102." *Id.* ¶38-40. Ellis itself defines "sporting event" as including "online sports commentary." *Id.* ¶38. And, media content includes broadcast programming "and any other content capable of being displayed by, presented to, … or interacted with, using user equipment 102." *Id.* ¶40. Also, Ellis teaches providing sports commentary in a data feed or over communications channel to a user. *Id*., ¶41. In my opinion, inclusion of commentary in the video and/or audio data being served to users according to the system of Herzog would have been more than obvious to a POSA. For example, commentary as taught by Ellis would have been supported as an audio/video channel in an HLS media stream. Herzog, ¶34 ("user may select, by way of keypad 120, specific audio and video data for listening and viewing"), ¶45 ("video and audio data may be multiple video/audio streams").

378. Herzog discloses a content provider (42) that provides a live stream of content, including video, to a solutions integrator: (44), who then provides the content to an end user device (45):

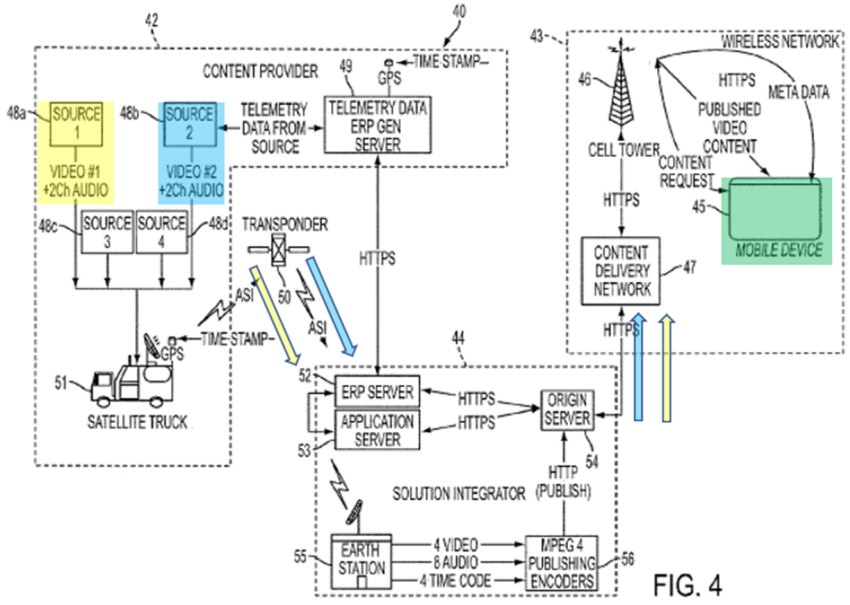


FIG. 4

379. Herzog discloses providing "multiple video/audio streams of race cars and drivers, as they are imaged at different locations on the race track:" Ex.1015 ¶45, Fig 4. The above system focuses on one type of live event, an Indy Car Race, but Herzog notes that it can be used for any type of live sporting event. *Id.* ¶¶45, 83. The system makes these video feeds available to "mobile stations" and other "similar users." *Id.* ¶19. Each user can then request video from "one or more channels of video/audio for viewing a live event." *Id.* ¶70. The user then "receives both the video/audio stream in an HTTPS Live Streaming (HLS)-based media streaming communications protocol." *Id.* ¶51. Thus, Herzog provides a system where a

plurality of different users can each receive a respective copy of a first broadcaster's live stream of digital content including first live sporting event video-based commentary from a broadcaster client device to a plurality of VCDs.

ii. **Element 23.1: "A) providing the first copy to the first viewer client device via a first video communication channel between a plurality of media sources and the first viewer client device;"**

380. In my opinion Herzog discloses this limitation.

381. Herzog discloses multiple MPEG 4 Publishing Encoders ("Encoders") as a plurality of media sources. Ex.1015 ¶48. In my opinion—just like in the '697—a POSA would have understood that these encoders receive all of the video and audio data from the live event, such as the Indy Car Race. *Id.* ¶¶46 (noting that any form of communication technologies may be used to send the video/audio data to MPEG Publishing Encoders), ¶49. The Encoders can "encode" the video and audio data and then the "encoded video is then published to the content delivery network 47 (for example) via IP utilizing [] HLS." *Id.* ¶¶49, 55. For example, Herzog provides the selected video stream from the Encoders to each user individually via "an HTTPS Live Streaming (HLS)-based media streaming communications protocol." Ex.1015 ¶51, Figs. 1, 4.

iii. **Element 23.2: "B) periodically retrieving, via the Internet and from an event information provider, first event information germane to a first live sporting event, wherein the first event information includes first score information for the first live sporting event; and"**

382. Herzog's ERP Server connects to the ERP Gen Server over the internet:

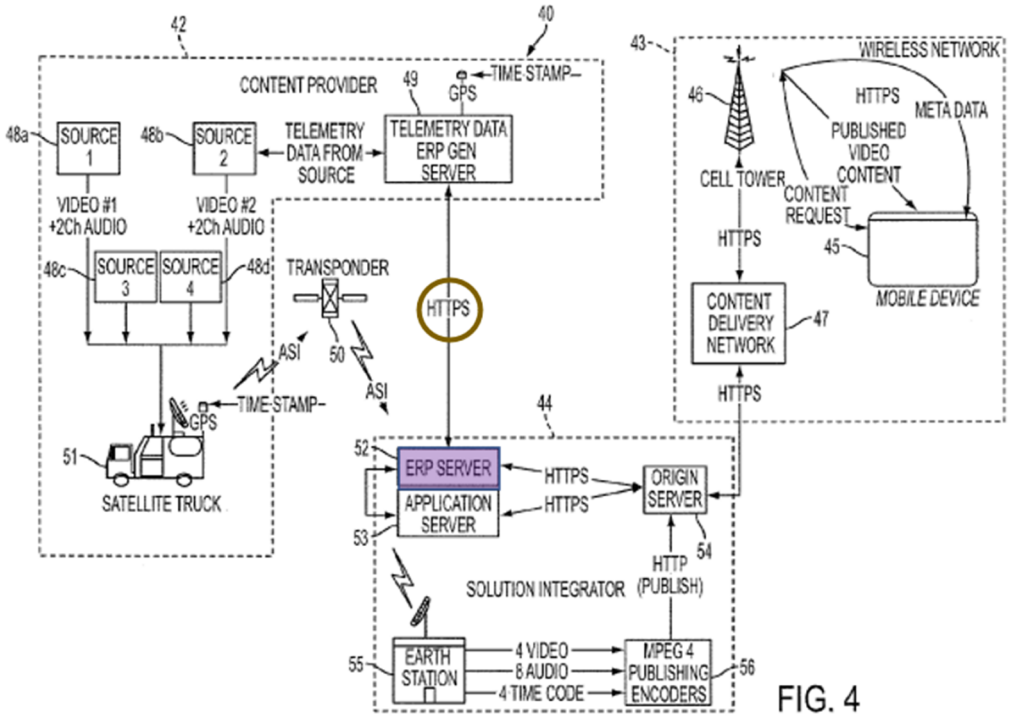


FIG. 4

Ex.1015 ¶¶45, 47, 50, Fig. 4 (annotated). The ERP Server is a generic server that retrieves meta-data from the ERP Gen Server. *Id.* ¶¶16, 47, 50. In my opinion, it contains the same components as the control server described in the '697: a processor, memory device, and communication interfaces. *Id.* ¶13, 85-86, Fig. 8. The ERP Server "acts as an intermediary between the ERP generator server and the origin server." Ex.1015 ¶50. The meta-data retrieved includes event information germane to the sporting event such as "the status of each car and driver in the race"

and may include "lap number, car speed and acceleration" but there may be as many as "50 different parameters that are available." *Id.* ¶45. The ERP Server retrieves this first score information via the internet and from an event information provider because it retrieves the information, *e.g.*, status of each car and driver in the race, in HTTPS format, using an internet communication protocol from an ERP Server. *Id.* ¶¶17, 50.

383. Herzog discloses that this metadata is "changing during the event." *Id*., ¶45. Thus, a POSA would have understood that the operation of the interconnected ERP Gen Server and ERP Server disclosed by Herzog over the course of the event results in the periodic retrieval of the metadata by the ERP server in accordance with HTTP protocol.

384. A POSA would have also understood that HTTP is generally a request-oriented protocol where receivers such as the ERP server need to explicitly ask for (pull) new data. To reduce latency using an HTTP connection, a POSA would have known to have Herzog's ERP server periodically check for the content availability as often as possible/reasonable. Alternatively, as discussed below, in my opinion it would have been an obvious design choice to use a periodic update to limit queries to the system and load on the components.

385. Ellis's SCS also 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 . . .”). Additionally, Ellis’s 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.” Ex.1016 ¶¶41-42, 142. The SCS periodically retrieves event information germane to all live sporting events, *e.g.*, score information because the SCS “provide[s] real-time sports scores and game summary information, … spread and wagering data, and any other sports-related information to user equipment 102.” *Id.* ¶41; *see id.* ¶¶72, 117. In my opinion, a POSA would have understood that this information is retrieved via the internet. *Id*. ¶¶37-38, 40, 142. Furthermore, a POSA would have understood that this was how real-time data was routinely transferred to users at the time of Herzog, Ellis and Spivey—including score updates and wagering information. Ex.1016 ¶¶42, 142. Additionally, in my opinion 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.

386. While a periodic time interval, e.g., every .1 seconds, is not mentioned in Herzog or Ellis, it is my opinion that a POSA would have understood that any system that provides real-time sports statistics, such as Herzog, would update the statistics on a periodic basis. 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, yet to limit queries to the system and load on the components. Too many requests in a small period of time may overload servers, decreasing responsiveness and without providing any real benefit. In my opinion, a POSA would have recognized the period of retrieval should be tailored to the nature of the event and/or the use case. For example, polling for scoring updates on a basketball game at a frequency of 1000 times a second may provide little benefit where the frequency of scoring is tied closely to a 24-second shot clock.

iv. **Element 23.3: "C) transmitting the first score information to at least the first viewer client device via a first event information communication channel between at least one first event socket of at least one socket server and the first viewer client device of the first plurality of viewer devices, wherein the at least one first event socket corresponds to the first event information germane to the first live sporting event."**

387. In my opinion, Herzog in view of Ellis and Spivey discloses a socket server communicatively coupled to the control server (which, as discussed above, is disclosed by Herzog's ERP Server). Herzog discloses providing event information over a separate connection to user equipment. Ex.1015 ¶¶48-52, 67, 70-77, Figs. 4, 6. That process includes the ERP Server transmitting data to the Origin Server:

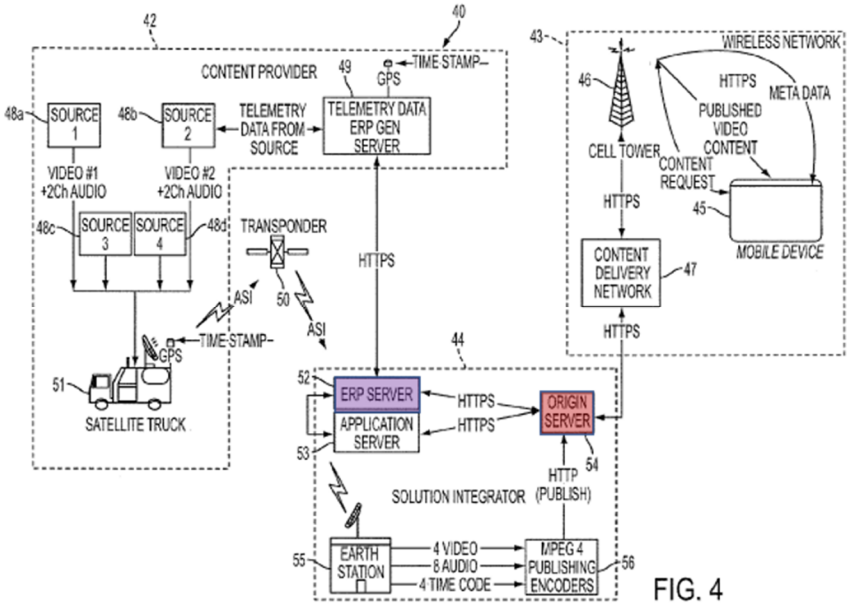


FIG. 4

Ex.1015 ¶¶50-51, 83, Fig. 4. But in a system of Spivey combined with Herzog and Ellis, it is my opinion that a POSA would have looked to Spivey to add additional backend architecture to organize the meta-data from multiple events and reduce latency in the real-time data delivery system for at least the same reasons set forth in §X.6.a.iv. I believe a POSA would have done this by incorporating Spivey's MQSD into at least Herzog's Origin Server.

388. Like the '697's socket server, Spivey's MQSD is a generic server that contains processors, memory, and communication components. Ex.1017 ¶¶35, 53, Figs. 3, 4.

389. Moreover, in my opinion a POSA would have understood that Spivey's MQSD carries out the same purpose as the '697's socket server. The MQSD "receive[s]" and "process[es] live data to provide live data" to "the subscribed

connected client devices substantially simultaneously." *Id.* ¶36; *see id.* ¶¶45 ("[MQS] device 310 may be a representative example of message queue server 120"), ¶54 (describing that 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:"

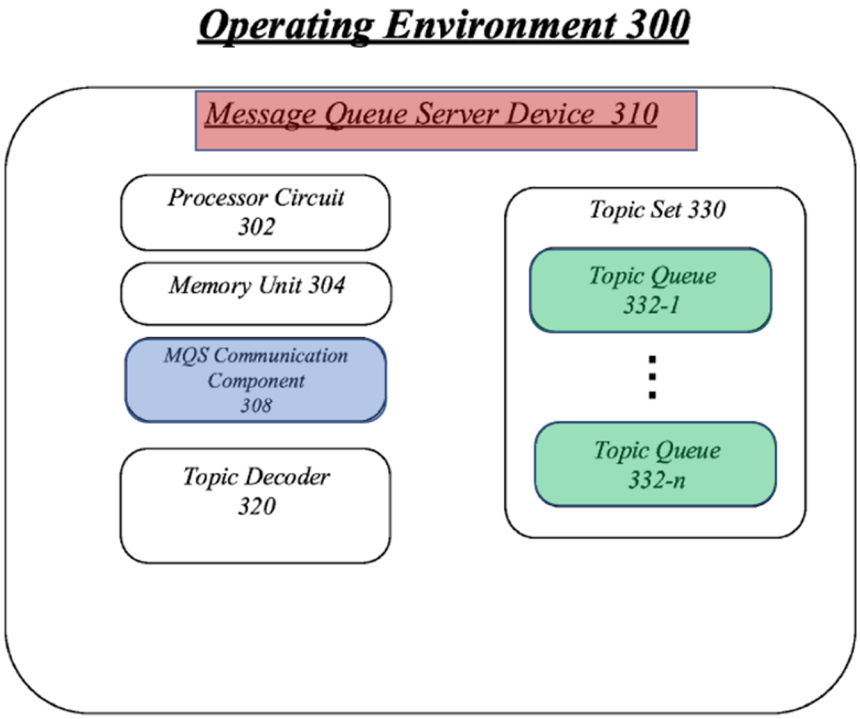


FIG. 3

Id. ¶55, Fig. 3 (annotated). MQSD contains the MQS communication component which would allow the MQSD to communicate with the ERP Server, as contemplated in Spivey. *See id.* ¶¶46 (disclosing that communication component 308 transmits and receives data), ¶54 (disclosing that communication component

408 is analogous to 308), ¶55 ("LDS communication component 418 … allows [LDSD] to … transmit the data payloads to [MQSD]."); Ex.1015, Fig. 4.

390. After receiving the data, 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. After sorting the data into the proper topic queues, the MQSD shares the information "substantially simultaneously" to "the subscribed connected client devices." *See id.* ¶96.

391. Spivey teaches that latency associated with the broadcast of event data, such as sports scores, to clients can be reduced by "using topic cues and open duplex communication connections." *Id.* ¶¶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."). As an example of such "open duplex communication connections" Spivey discloses that communications between a server and a client 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"); *see also*, *Id.* ¶¶ 43, 61, 74, and 92; claim 8. Thus, it would also have been obvious to a POSA in view of these teachings of Spivey 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. As explained above, a POSA would have recognized that doing so would further the express goals of Spivey with respect to reduction of latency in such transmissions. In my opinion, a POSA would have had a reasonable expectation of success implementing this web socket functionality in the MQSD because this would have involved nothing more than a combination of well-known, compatible elements to achieve a predictable result.

(A) First event socket corresponds to the first event information germane to the first live sporting event

392. For at least the same reasons set forth in §X.6.a.iv., Herzog as modified by Spivey in view of Ellis satisfies this aspect of the claimed invention. Spivey often discusses a connection between the client device and the system using a live data access server ("LDAS"), however, Spivey also discloses and teaches that the message queue server device may be "accessible to clients over the network directly." Ex.1017 ¶¶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 disclose transmitting first event information on"), 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. Also, the system does not require an LDAS, only that it "may include a[n] [LDAS]." *Id.* ¶38. Thus, it is my opinion that a POSA would have

understood Spivey to disclose and teach a direct connection between the client devices and MQSD. In my opinion, a POSA would have used this implementation because reducing the amount of server transfers would reduce latency, which is Spivey's main purpose. Furthermore, it was a well-known alternative to a POSA to implement the functionality of multiple servers as a single server. Ex.1016, ¶¶4-7.

393. In my opinion, a POSA would have understood that 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. *Id.* It is my opinion that 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. *See* Ex.1006 at 20:45-61, 21:53-22:6; Ex.1029, pp. 14-20; Ex1030; §IV.C.3., *supra*.

394. In the same way, 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 then 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.

395. 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. *See 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:

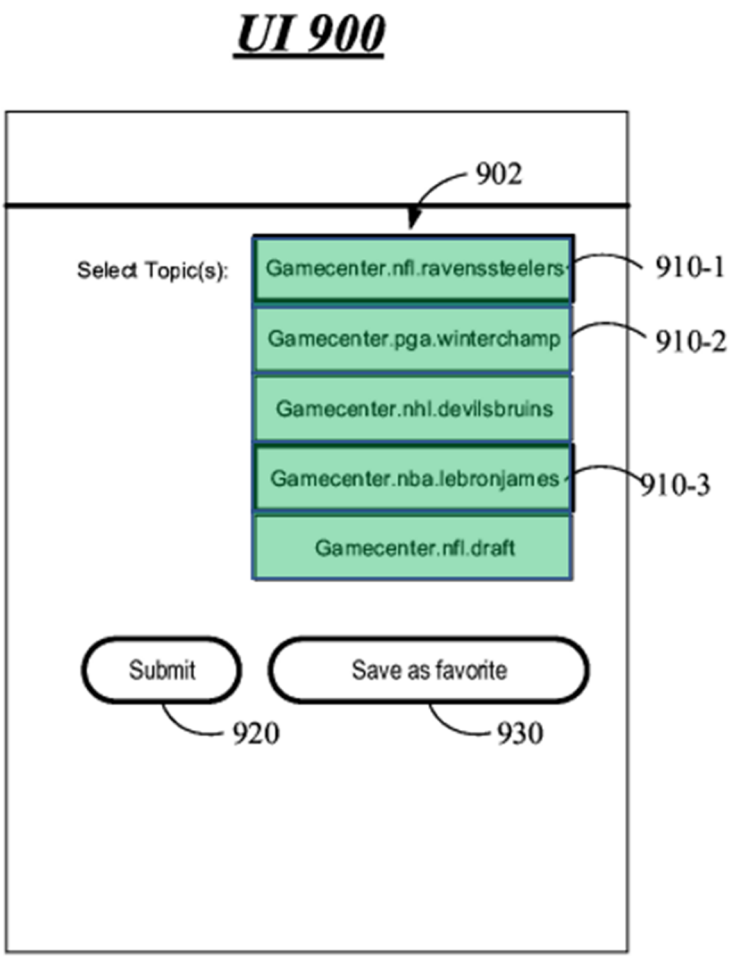


FIG. 9

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, it is my opinion that a POSA would have understood that Spivey's topic queues disclose the '697's first event sockets corresponding to the first event information germane to the first live sporting event.

396. Alternatively, it would also have been obvious to a POSA in view of these teachings of Spivey 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 in §IX.A.1, IX.F and §X.A.1.a.vi.

(B) Transmitting the first score information to at least the first VCD

397. For at least the same reasons set forth in §X.6.a.iv., Herzog as modified by Spivey in view of Ellis satisfies this aspect of the claimed invention. In my opinion, Herzog discloses a first VCD. *See* §X.A.8.a.i-iii., *supra.* In my opinion, a POSA would have understood that both Herzog and Spivey further disclose transmitting the first score information to the first VCD. 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, 48 ("data payload may include, for example, a score, a play, an error, an injury, a player at-bat, and so forth"). In Herzog, the Origin Server may be configured to transmit "the meta-data to the

content delivery network 47," which includes car position, and the user receives "in an HTTPS communications protocol." Ex.1015 ¶¶45, 51.

(C) Transmitting via a first event information 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

398. For at least the same reasons set forth in §X.6.a.iv., Herzog as modified by Spivey in view of Ellis satisfies this aspect of the claimed invention. As previously noted, Herzog discloses at least a first VCD. *See* §X.A.8.a.i-ii. However, to the extent that the claim is interpreted as reciting "***a*** *first plurality of viewer client devices*," Herzog discloses providing the received copies to multiple different users with the option to select any of the available video feeds. Ex.1015 ¶¶19, 49, 70, FIG. 1. As would have been obvious to a POSA, in a content distribution scheme such as Herzog that provides at least four different video streams, two or more users would select a first stream to view, and two or more different users would select a second stream to view, *e.g.*, the broadcast from their favorite driver's car. Ex.1015 ¶¶46, 70. In my opinion, a POSA would have understood this to at least be a first plurality of VCDs containing a first VCD.

399. Herzog provides the selected video stream to each user individually via "an HTTPS Live Streaming (HLS)-based media streaming communications protocol." Ex.1015 ¶51, Figs. 1, 4. Thus, it is my opinion that a POSA would have

understood that each user device receives their own copy of the video stream. This would create a plurality of copies being distributed, where the first copy is sent to the first user, and so on. Thus, in my opinion a POSA would have understood that each user would receive an individual copy of the broadcaster's live stream of digital content.

400. Herzog discloses providing individual copies of multiple different live streams to multiple groups of client devices. Ex.1015 ¶¶46, 49, 70. Herzog further discloses providing video content over a video Internet communication channel via HTTPS Live Streaming, *i.e.*, an internet connection. Ex.1015 ¶51. Thus, it is my opinion that Herzog discloses that each user receives an individual copy from the Encoders over a video internet communication path that is unique to each user device. *Id.* ¶¶2 (noting that user devices "are used mainly for viewing published content from the Internet, such as video."), 51.

401. In my opinion, Herzog in view of Spivey and Ellis 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 Internet." *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, Herzog discloses providing event information over a separate communication path than the video signal where there is "a unique video and meta-data source that do not follow the same path of transmission:"

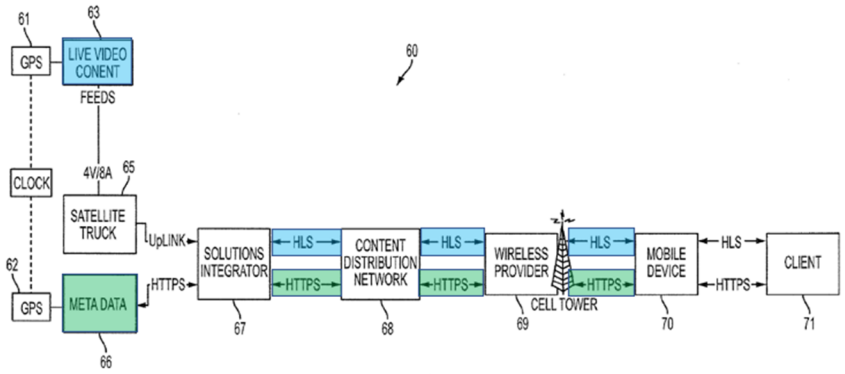


FIG. 5

Ex.1015 ¶47-51, 83, Figs. 1, 4, 5 (annotated) 6. Thus, it is my opinion that a POSA would have understood Spivey discloses transmitting first score information via a separate internet connection. In my opinion, a POSA would have known, and considered obvious, that the websocket client-server communication protocol employed by Spivey was an alternative to Herzog's HTTP polling. §IV.C

402. Further, in my opinion this information is provided via a connection between a topic queue and the first user equipment device. Users can access the MQSD using a URL. *See* Ex.1017 ¶¶81-82. Then, the user can access the specific topic queue she wishes to receive information from by using a text string, *e.g.*, a topic label or Event ID, associated with a topic queue. *Id.* ¶179, Fig. 9 ("Gamecenter.nfl.ravenssteelers."). Thus, it is my opinion that a POSA would have

understood that Herzog in view of Ellis and Spivey discloses distributing event information from a socket to a user device over an internet connection. Further, I believe that 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.

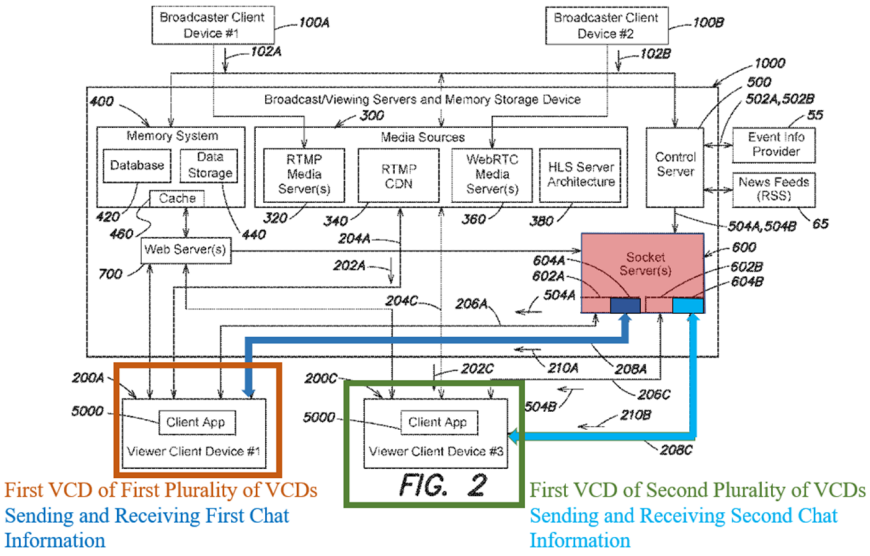
b. Dependent Claim 24

i. **Elements 24.0-24.1: "The method of claim 23, further comprising: D) transmitting and receiving first chat information regarding the first live sporting event video-based commentary included in the first broadcaster's live stream of digital content via at least one first chat/system event Internet communication channel between at least one first chat/system event socket of the at least one socket server and the first viewer client device of the first plurality of viewer client devices, wherein the at least one first chat/system event socket corresponds to the first chat information regarding the first live sporting event video-based commentary."**

403. In my opinion, Herzog in view of Ellis and Spivey discloses these elements. Herzog in view of Ellis and Spivey discloses transmitting information via individual communication channels between a socket and a first VCD of a first plurality of VCDs who are watching the same event. §X.A.8.a *supra*.

404. The '697 explains that a chat/system event communication channel may be any generic internet communication channel. Ex.1006, 5:29-36. These

elements only further require that the system "transmits and receives" first "chat information regarding" the first "live sporting event video-based commentary included in the first stream of digital content" over another separate, individual connection between a first "chat/system event socket" and a first VCD:



Ex.1006, 22:7-39, Claim 24, Fig. 2. In sum, it is my opinion that a POSA would have understood that these elements require that chat information is transmitted and received over a communication channel from another socket that is dedicated for chat/system information.

405. The '697 describes chat information as communications "between a given broadcaster … and corresponding viewers … of chat messages." Ex.1006, 20:12-17. Further, "viewers of the same live event may" view different video feeds of the same live event and "participate in different chat information exchanges … and all share the same event information in a synchronized manner." *Id.,* 23:8-19.

Additionally, the '697's explanation of chat/system event sockets indicate that they are the same as the other sockets, *i.e.*, a communication endpoint on a server that is uniquely addressable, *e.g.*, by using an identifier (StreamID) that may be contained, *e.g.*, in a URL. *Id*., 20:45-61, 21:53-22:6. The only difference is that these sockets store and send chat messages, not "event data." *See id.,* 23:46-24:5.

406. Herzog as modified by Ellis and Spivey render these elements obvious for lease the same reasons set forth above at §X.6.b.i. Ellis discloses chat information because users "may also leave feedback regarding the commentary that other users in the media system may view when they access the same commentary." Ex.1016 ¶132. Thus, a POSA would have understood that Herzog in view of Ellis and Spivey discloses a system that transmits and receives chat information regarding the live stream of digital content. *Id.* ¶126 ("To create and send a message (e.g., instant message or email message), the user may select send message option.").

407. Ellis, consistent with Herzog's separate meta-data stream, also discloses providing this event data and chat information on a separate channel to users. Ex.1015 ¶¶49-53, 59-61, Fig. 5; Ex.1016 ¶¶39-41, Fig. 1. 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. Consistent with Herzog, Ellis provides that in "some embodiments, this sports-related information is provided to user equipment

102 as a plurality of real-time data feeds." Ex.1016 ¶41. Thus, A POSA would have understood that Herzog in view Ellis of Spivey discloses transmitting via at least one first chat/system event Internet communication channel between at least one first chat/system event socket of the at least one socket server and the first VCD.

408. Herzog in view of Ellis and Spivey discloses a socket specifically designated for chat information. Spivey utilizes topic queues to organize "data payloads" that "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." 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. For example, if there were two topics "at the same hierarchical level" the data payloads placed in "'gamecenter.nfl.ravenssteelers.statistics' would not be placed in" another topic queue "gamecenter.nfl.ravenssteelers.plays." *Id*. In sum, the statistics would be provided separately from the other information, e.g., chat information. Thus, a POSA would have understood Spivey to teach a first chat/system event socket (*e.g.*, "gamecenter.nfl.ravenssteelers.chat") corresponding to the first chat information.

9. '697 Ground 9: Claim 26 is Obvious Over Herzog in View of Spivey, Ellis, Kellicker, and Phillips

409. For the reasons stated in Section IX.G, above, it is my opinion that a POSA would have been motivated to combine the teachings of Herzog, Spivey, Ellis, Kellicker, and Phillips to arrive at the claimed invention with a reasonable expectation of success.

i. **Elements 26.0-26.2: "The method of claim 24, wherein: the plurality of media sources comprises: at least one of: at least one real-time messaging protocol (RTMP) media server to receive the first broadcaster's live stream of digital content; and at least one web real time communication (WebRTC) media server to receive the first broadcaster's live stream of digital content; and"**

410. In my opinion, Herzog in view of Ellis, Spivey and Kellicker teaches this element, for at least the same reasons set forth in §X.A.7.a.i..

411. As explained in §IX.B above, Kellicker discloses a server (120/1250) that receives incoming live media streams (e.g., live incoming media stream (1220)), processes the incoming media streams and serves such media streams to playback devices in real-time. Ex.1049, ¶¶100-101, 106. Kellicker discloses that this server can be implemented through a plurality of such 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"), ¶24. Thus, Kellicker

discloses a plurality of media sources in the form of real-time streaming media servers.

412. Kellicker also teaches that the server(s) may support various streaming protocols including, RTMP, and WebRTC. *Id*., ¶106. ("computing device 1250 may also support real time messaging protocol (***RTMP***) (and variants thereof), real-time streaming protocol (RTSP), real-time transport protocol (RTTP), web real-time communication (***WebRTC***), and MPEG-2 transport stream (MPEG-TS)"), ¶ 24 (disclosing transcoders/transmuxers for transforming media streams). Thus, Kellicker discloses media sources in the form of a real-time RTMP media server and/or a real-time WebRTC server.

413. Kellicker teaches providing its server(s) with the capability to "enable the computing device 1250 to process data in accordance with multiple coding technologies and protocols." *Id*., ¶103-104. In my opinion, it would have been obvious to a POSA to have implemented the media sources (i.e., "encoders" 56) of Herzog, at least in part, in the form of a real-time RTMP media server and/or a real-time WebRTC server. In my opinion, a POSA would have been motivated to do so in order to provide the content source of Herzog with the processing and transmission ability to conform with these well-known and widely utilized protocols. *Id*., ¶¶103-106.

414. To the extent required by the claims, in my opinion it would have also been obvious to a POSA to have modified Herzog to implement the media sources 56 in the form of multiple servers, including a RTMP server ***and*** a WebRTC server. A POSA would have recognized that the use of multiple servers, as taught by Kellicker, would have provided the system with even more flexibility, capacity and adaptability to handle multiple different types of incoming media and comply with multiple types of protocols. *Id*., ¶98.

ii. **Element 26.3: "an HTTP live streaming (HLS) caching server architecture communicatively coupled to at least one of the at least one RTMP media server and the at least one WebRTC media server,"**

415. In my opinion, Herzog in view of Ellis, Spivey and Kellicker teaches this element, for at least the same reasons set forth in §X.A.7.a.ii.

416. Kellicker teaches that the at least one RTMP media server and/or the at least one WebRTC media server are communicatively coupled to one or more other servers 1280. Ex.1049, ¶102. For example, 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., an edge server), etc. *Id*.; FIG. 12. Kellicker also explains that the data transmitted to the servers 1280 includes "media

streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof)." *Id*.

417. Kellicker further discloses that the media streams received by the one or more other servers 1280 "may be associated with the same encoding format and transmission protocol or may be associated with different encoding formats and transmission protocols." *Id*. and ¶24 (disclosing transcoders/transmuxers for transforming media streams). Kellicker also specifically teaches providing the real-time streaming server(s) with the HLS streaming protocol ability. *Id*. at ¶105-106. In my opinion, it would have been obvious to a POSA to implement such HLS conversion capability in the form of the one or more other servers 1280, such as the media/relay server 1281, for the Herzog's encoders 56 (media sources). A POSA would have been motivated to provide HTTP live streaming (HLS) media streams because, as was well-known to a POSA, HLS "provides a reliable, cost-effective means of delivering continuous and long-form video over the Internet." Ex.1052, 2 (§1).

418. 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." *Id*. at ¶102. In my opinion, a POSA would have understood a CDN to be a caching architecture. Ex.1052, 2 (§1) ("[HLS] offers compatibility with large-scale HTTP caching infrastructure to support delivery to large audiences"); Ex.1055

Lepeska, 7:19-23 ("[a] content delivery network or content distribution network (CDN) is traditionally known as a system of computer containing copies of data placed at various nodes of a network, such that access to the cached data is improved by increasing bandwidth and reducing access latency"). To a POSA, the fundamental purpose of a CDN is to provide caching capability in order to improve responsiveness to client device requests for served data (in this case streamed media. *See also,* Phillips (Ex.1050). Thus, in my opinion, a POSA would have understood that Kellicker teaches an HLS caching server architecture communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server.

iii. **Element 26.4-26.5: "wherein the HLS caching server architecture comprises: an HLS mother server communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server to receive, from the at least one of the at least one RTMP media server and the at least one WebRTC media server, a first HLS file suite frepresenting the first broadcaster's live stream of digital content;**

iv. **a plurality of HLS child servers communicatively coupled to the HLS mother server and an HLS content delivery network (HLS CDN);"**

419. In my opinion, Herzog in view of Ellis, Spivey, Kellicker, and Phillips teaches this element for at least the same reasons explained at X.A.7.a.iii.-iv.

420. As explained above at §X.A.2., Kellicker teaches an HTTP live streaming (HLS) caching server CDN architecture including a CDN communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server. It would have been obvious to a POSA to have provided this HLS caching server CDN 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." Ex.1050, ¶31. Consistent with the teachings of Phillips, a POSA would have been motivated to provide this server architecture in order to provide "for high-performance streaming . . . to subscribers using one or more Internet-based infrastructures." Id., ¶30. For example, Phillips teaches that 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 implement Phillips' 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 . . .").

421. 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, coupled to an HLS CDN architecture (1282), and from the CDN to a client/user device (1270):

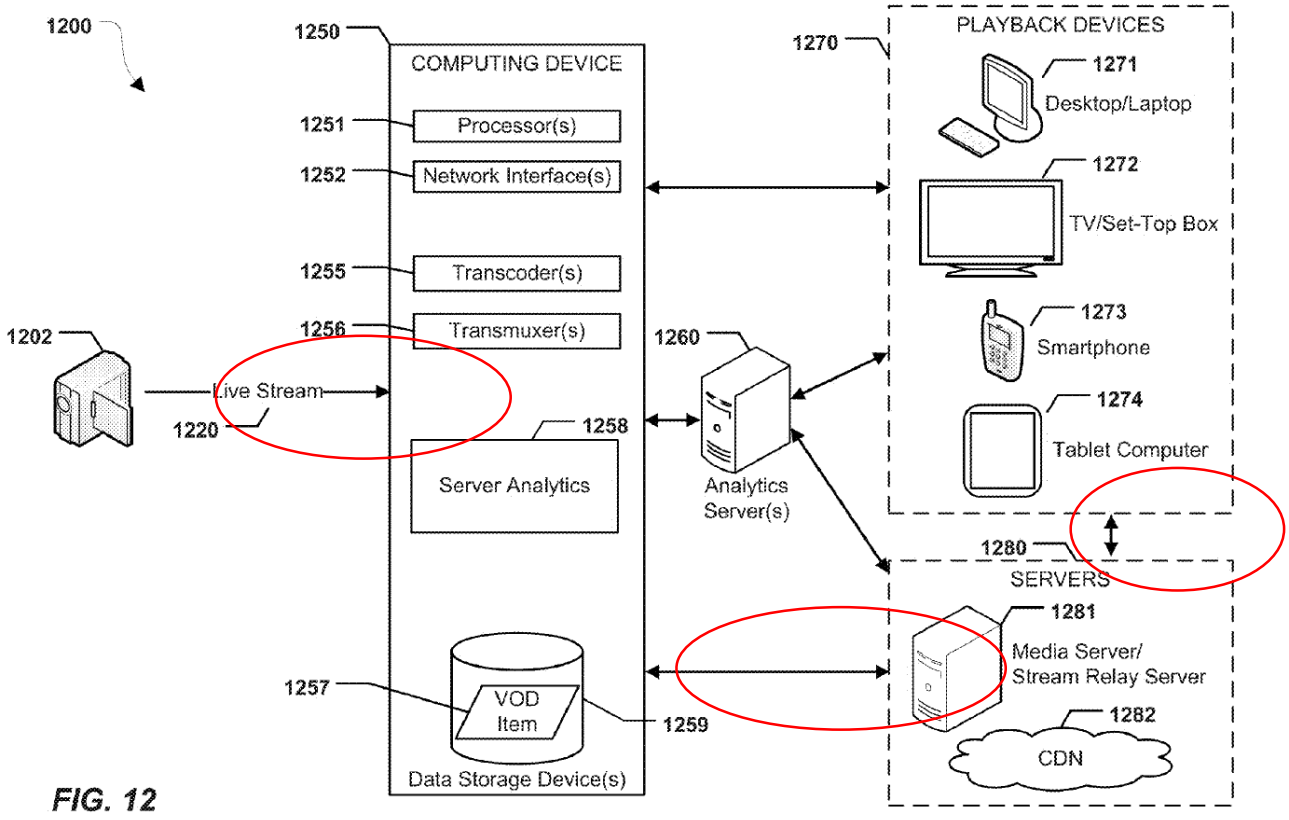


FIG. 12

Ex.1049, Fig.12 (annotated).

422. In particular, Kellicker discloses that the media server(s) 1250 is configured to transmit data to "a server of a content distribution network (CDN) 1282." Id., ¶102. Further, Kellicker discloses that, "[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.

423. In similar fashion, Phillips teaches that content that is 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 that 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 contents and/or nature of the files being handled by the system could be 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.1050, §§8.4-8.6. As can be seen, Phillips' 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.

424. With respect to the "HLS file suite" this represents nothing more than conventional HLS file transfer protocol, as acknowledged in the '697. Ex.1006, 38:47-55. 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.1050, §§8.1-8.7. By delivering content from Kellicker's server 1250 to a Phillips-type CDN, the HLS caching server architecture comprises: an HLS mother server (e.g., national server, regional servers) which are communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server.

425. Hence, 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 POSA would have understood that 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. As explained in the '697, an "HLS file suite" represents nothing more than the contents of a conventional HLS file transfer protocol. Ex.1006, 38:47–55. In my opinion, 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.1052, 27-29 (§ 8.1-8.7).

426. Alternatively, and to the extent that the nature or content of the data could be considered as limiting the claimed system, 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.1050, §§8.1-8.7. For example, as explained in Kellicker, this HLS caching server architecture receives a portion of the first HLS file suite (video stream) representing the first live stream of digital content. Ex.1049, ¶102 ("data transmitted to the playback devices 1270 and to the servers 1280 includes media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof"). Further, Kellicker discloses that "first HTTP port(s) may be used to communicate a master rendition list and media chunks . . .," each of which is a portion of the HLS file suite. Ex.1049, ¶110; *see also* Ex.1006, 38:47–55.

427. Similarly, Phillips teaches that the first HLS file suite is received (media chunks and manifest/playlist). For example, Phillips teaches that content that is requested for delivery by the CDN may include a received "manifest" that is made available to a "content delivery node" (e.g., "mother" or "child" server) of the CDN server architecture. Ex.1050, 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"). And, to the extent that 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 (e.g., a "child" server). Ex.1050, Abstract, ¶31, Fig.

428. As another example, Phillips explains that the media portions of the HLS file suite are received by the CDN architecture. Ex.1050, FIG. 12 ("Pull Missing Segments for ABR Assets", which are media chunks of the HLS file suite), ¶9.

v. **Element 26.6: "A) comprises providing the first copy to the first viewer client device via the first video communication channel between the HLS CDN and the first viewer client device, wherein the first copy is based on the first HLS file suite received by the HLS mother server."**

429. This element is satisfied for at least the same reasons set forth at §X.A.7.a.v. As explained above in §X.A.8.a.ii, Herzog as modified by Ellis and Spivey establishes the first video communication channel to provide the first copy of digital content to the first viewer client device. As further explained at §§X.A.2.a.iii-iv, X.A.3.a.ii-iii, and X.A.9.a.i-iv Herzog as modified by Ellis, Spivey, Kellicker and Phillips establishes that the first video Internet communication channels are provided between the first viewer client device and the HLS CDN. *See also* Ex.1050, ¶33 ("Each local edge server is operable to serve a plurality of subscribers in a serving area for delivering content to associated end stations or client devices, as illustrated by an example client device or user equipment (UE) device 105 that may for consume or access streaming content delivered via CDN 100 in any

type or number of access technologies including broadband access via wired and/or wireless (radio) communications."), ¶28 ("client devices may also access or consume content/services provided over… packet-switched wide area public network such as the Internet via suitable service provider access networks.").

430. As explained at §X.A.3.a.i.-ii and X.A.9.i.-iv., Herzog as modified by Ellis, 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 as required by the HLS specification and the CDN distribution process.

B. The '687

1. '687 Ground 1: Claim 1 is Obvious Over Ellis in View of Spivey

431. For the reasons stated in §IX.A. above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis and Spivey with a reasonable expectation of success to arrive at the claimed invention.

a. Independent Claim 1

i. **Element 1.0: "A system for providing, to a first plurality of viewer client devices, a first plurality of copies of a first live stream of digital content relating to a first live sporting event, and for providing, to a second plurality of viewer client devices, a second plurality of copies of a second live stream of digital content relating to the first live sporting event or a second live sporting event, the system comprising:"**

432. To the extent the preamble is limiting, it is my opinion that Ellis discloses this element. *See* Ex.1016, Abstract. Ellis discloses a system for providing a first/second plurality of copies of a first/second live stream of digital content including first/second 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. Figure 2B, below, shows live sporting events available at the same time to any number of users.

220

222 Tuner	224 Primary Source ID
1	<sourceid_2>
2	<sourceid_1>
3	<sourceid_8>
4	<sourceid_4>

230

232 Source ID	234 Type	236 Profile Score
<sourceid_1>	Game	90
<sourceid_2>	Game	95
<sourceid_3>	Commentary	45
<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 those 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.*

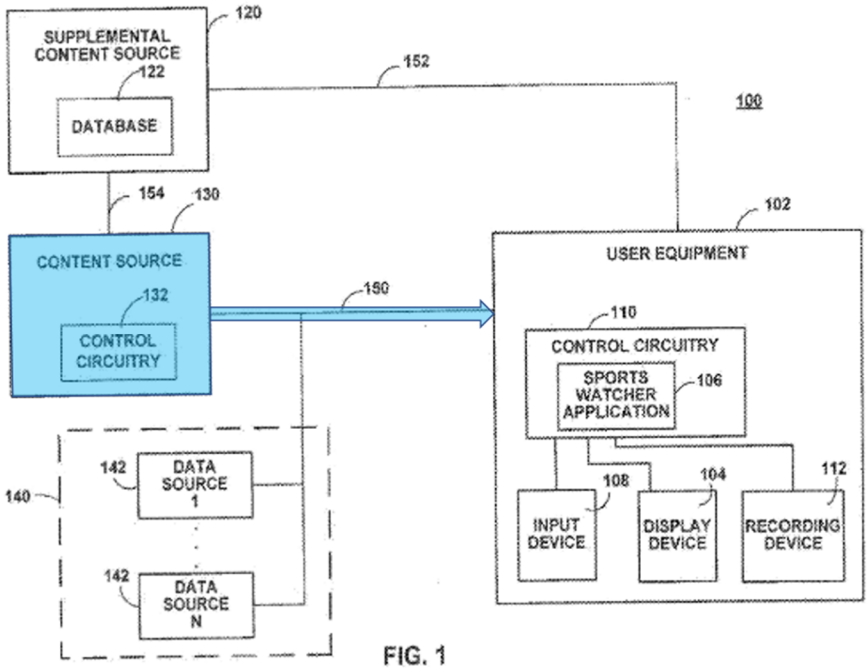
433. Ellis discloses multiple broadcasters providing copies of each broadcaster's live stream of digital content including content relating to the live

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. For example, Ellis's system broadcasts digital streams from the content source, which can be an "[i]nternet or network media/webserver." 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. *Id.* ¶45, 74. The communication path can be any "suitable wired or wireless path." *Id.* "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. In my opinion, a POSA would have understood Ellis' distribution system provides for a plurality of broadcaster devices sending, a plurality of copies of multiple live streams of digital content to a plurality of viewers.

ii. **Element 1.1: "A) a plurality of media sources to:"**

434. In my opinion, Ellis discloses content sources, which are "any suitable source of media content" such as an "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 serve to receive the first/second live stream of digital content and distribute video content to users

via communications paths. *See id.* ¶¶39-41, 45, 47. Figure 1 shows a content source distributing video content to users over communication path 150:



Ex.1016 ¶¶39-41, Fig. 1 (annotated). In my opinion, Ellis also discloses multiple content sources, *i.e.*, a plurality. *Id.* ¶39 ("[T]here may be numerous instances of content source 130 …"), 74. As shown in Fig. 1 above, Ellis also discloses that users can be connected with a plurality of data sources 140. *Id.* Moreover, as explained below, it is my opinion that Ellis's content sources perform the same functions as the media sources as recited in elements 1.2-1.4. Thus, a POSA would have understood that Ellis discloses the recited element.

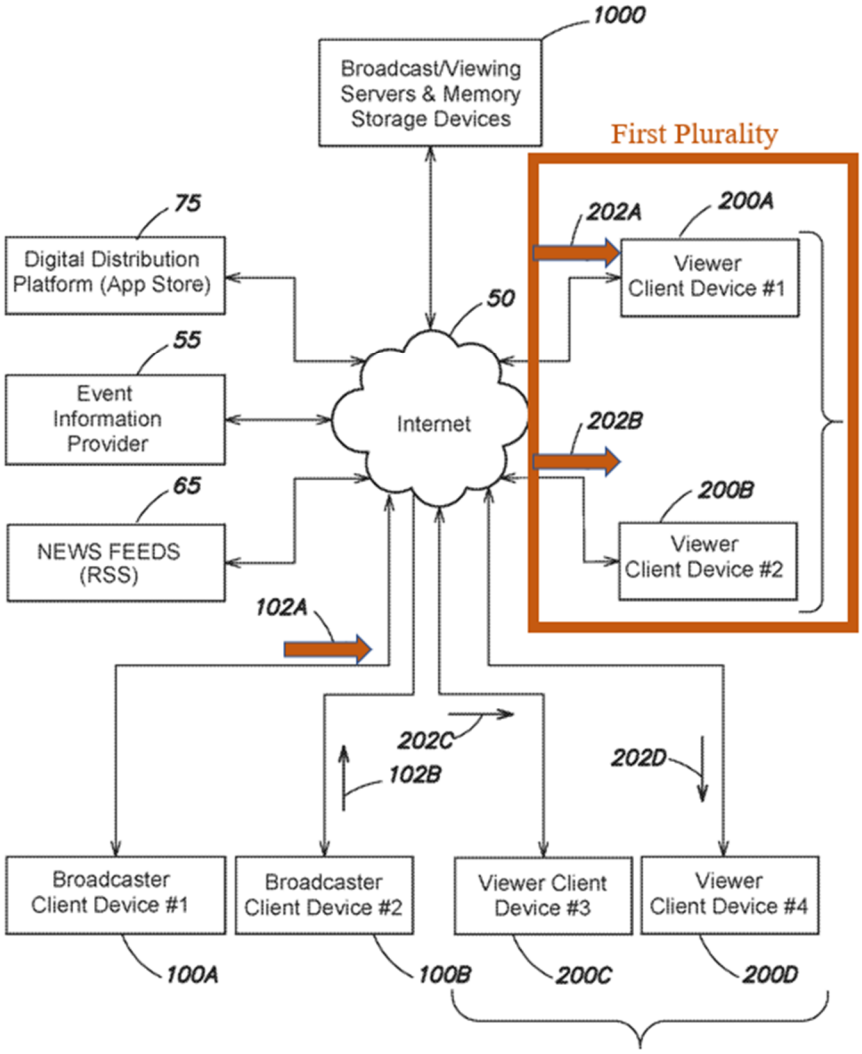
iii. **Element 1.2: "receive the first live stream of digital content and the second live stream of digital content;"**

435. In my opinion, Ellis discloses that the content source "may be any suitable source of media content, such as, for example a cable system headend, satellite television distribution facility, 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, in my opinion Ellis discloses content sources that receive multiple live streams of digital content and distribute that content to user equipment.

iv. **Elements 1.3-1.4: "provide a first copy of the first plurality of copies to a first viewer client device of the first plurality of viewer client devices via a first video Internet communication channel between the plurality of media sources and the first viewer client device of the first plurality of viewer client devices; and provide a first copy of the second plurality of copies to a first viewer client device of the second plurality of viewer client devices via a second video Internet communication channel between the plurality of media sources and the first viewer client device of the second plurality of viewer client devices;"**

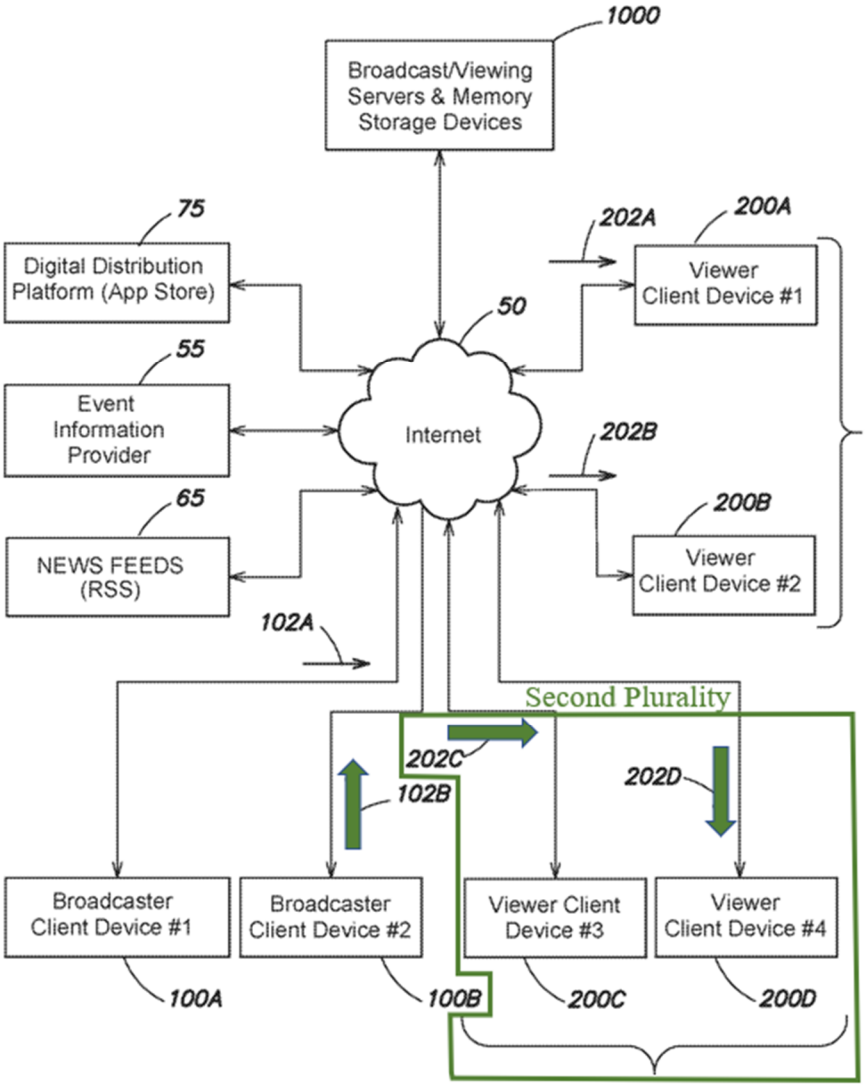
436. The '687's Figure 1A gives an example of the "first viewer client device [("VCD")] of the first plurality of viewer client devices." There, a first VCD

means one user, 200A, of the first group that receives individual copies 202A, 202B of the first live stream of digital content 102A:



Ex.1003, 9:25-50, Fig. 1A (explaining that Figure 1A depicts providing "a first [VCD] (200A)" of "a first plurality of [VCDs] (200A, 200B)" that receive "copies (202A, 202B) of a first broadcaster's live stream of digital content (102A)."). Likewise, "the first viewer client device of the second plurality of viewer client

devices" simply means one user of the second group of VCDs that receives individual copies of a second live stream of digital content:

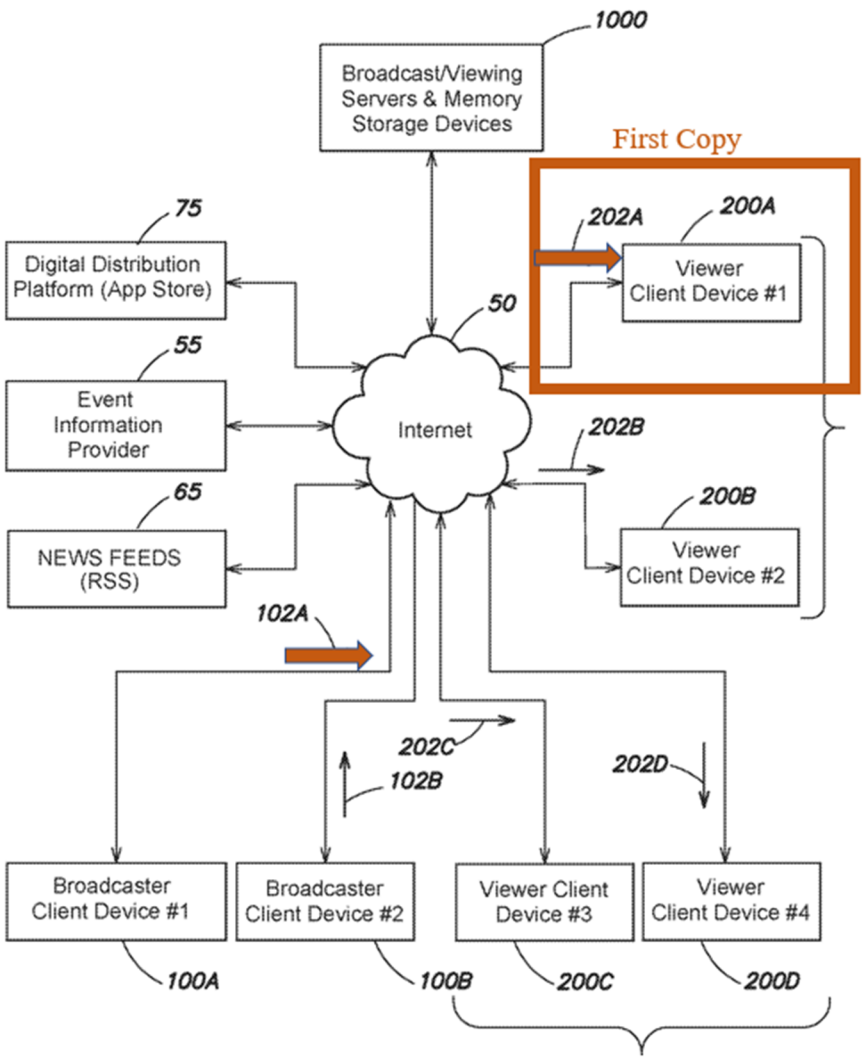


Id., 9:25-50(explaining that "a first [VCD] (200C)" of "a second plurality of [VCDs] (200C, 200D)" receive "copies (202C, 202D) of a second broadcaster's live stream of digital content (102B)."), Fig. 1A.

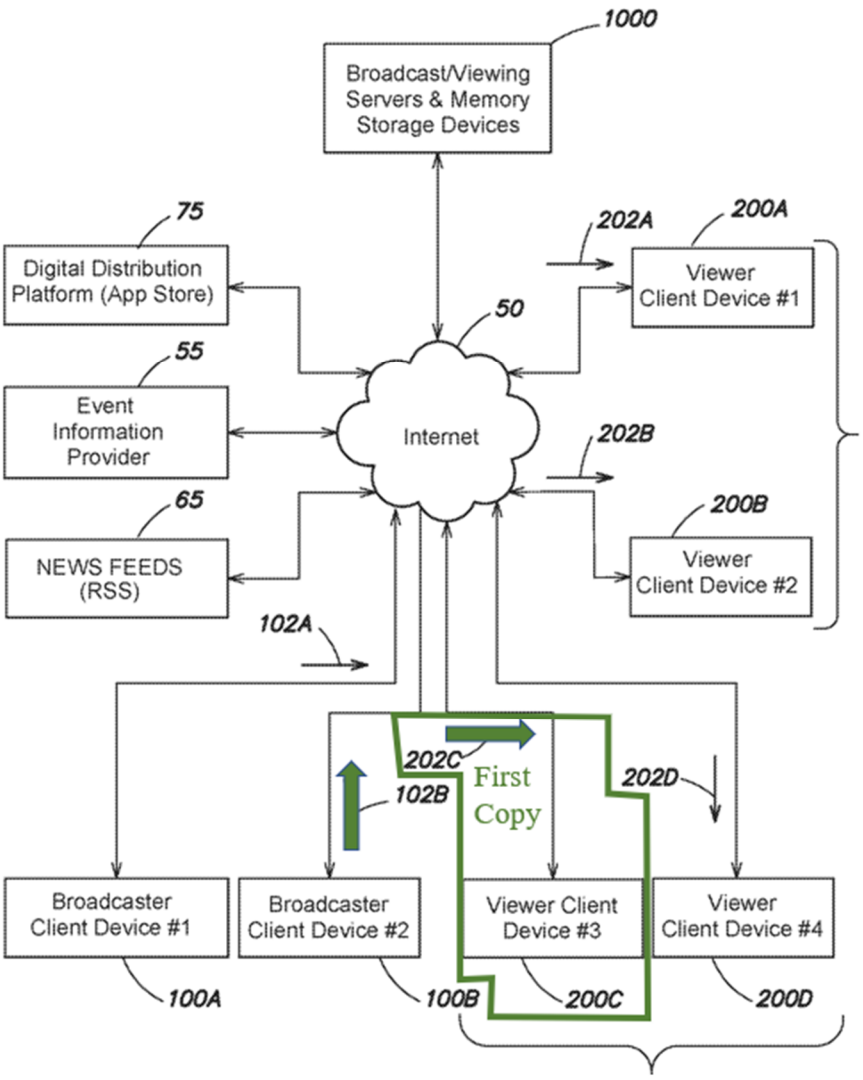
437. In my opinion, Ellis discloses a first VCD of a first/second plurality of VCDs because Ellis discloses an "interactive media system" where "any suitable

number of users" may "select any game from the listing" of "available games" to "watch the selected game." Ex.1016 ¶94. For example, a first user in a first group of users (the first plurality) can choose to watch the Yankees broadcast from the YES network. *Id.* ¶55. Meanwhile, a first user of a second group 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. Thus, it is my opinion that Ellis discloses multiple first users of multiple different pluralities of VCDs.

438. Further, as shown in Figure 1 below, the '687 explains that the first copy of "the first plurality of copies" means a single copy provided to one VCD, where another VCD receives a separate copy over a separate connection:



Ex.1003, 9:25-50(explaining that VCD 200A receives a "first copy (202A)" of "a first plurality of copies (202A, 202B) of a first broadcaster's live stream of digital content (102A)"), Fig. 1. Likewise, the first copy of "the second plurality of copies" means a single copy provided to one VCD, where another VCD receives a separate copy over a separate connection:



Id., 9:25-10:12 (noting that the system provides "a first copy (202C)" of "a second plurality of copies (202C, 202D) of a second broadcaster's live stream of digital content (102B)."), Fig. 1.

439. In my opinion, 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. Ellis's content source, an internet server, provides the live stream over a connection path, which can be any "suitable wired or wireless path" including an internet connection. Ex.1016 ¶¶39, 45. Thus, it is my opinion that each user would receive an individual copy of the live stream of digital content.

440. In my opinion, Ellis additionally discloses providing individual copies of multiple different live streams to multiple groups of client devices. Ellis further discloses providing digital content or video content over a first/second video Internet communication channel. As an example of an "Internet communication channel," the '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 to user equipment, over a communication path. Ex.1016 ¶39, 74. The content source can be an "internet or network media/web server" and the user equipment "may include computer equipment." *Id.* ¶¶40, 48. The communication path can be "for example, a satellite path, a fiber-optic path, a cable path, or any other suitable wired or wireless path." *Id.* ¶45. 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, in my opinion

Ellis discloses that each user receives a copy from the content source over a video internet communication channel/path that is unique to each user device. *Id*.

v. Element 1.5: "B) a control server to periodically retrieve, via the Internet, first event information germane to the first live sporting event; and"

441. In my opinion, Ellis and Spivey both disclose this element. The '687 explains that the control server is a generic server. Ex.1003, 55:15-25. The control server retrieves "event information" from an "event information provider." *Id.* 21:46-62. The control server then transmits the event information to the socket server. *Id.* 19:51-55, 21:47-58. 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.

442. In my opinion, 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." Ex.1016 ¶¶41, 42, 142. The SCS periodically retrieves, via the internet and from an event information provider, first event information germane to the first/second live sporting events, including first/second score information for the first live sporting event because the SCS "provide[s] real-

time sports scores and game summary information, … spread and wagering data, and any other sports-related information to user equipment 102." *Id.* ¶41; *see id.* ¶¶72, 117. In my opinion, a POSA would have understood that this information is retrieved via the internet. *Id*. ¶¶37-38, 40, 142. Furthermore, a POSA would have understood that this was how real-time data was routinely transferred to users at the time of Ellis and Spivey—including score updates and wagering information. Ex.1016 ¶¶42, 142. Additionally, in my opinion, 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. Ex. 1024, 408-410.

443. While a periodic time interval is not mentioned in Ellis, it is my opinion that 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. This is especially true given that no specific time interval is specified by the claim.

444. 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, yet to limit queries to the system and load on the components. Too many requests in a small period of time may overload servers, decreasing responsiveness and without providing any real benefit. In my opinion, a POSA would have recognized the

period of retrieval should be tailored to the nature of the event and/or the use case. For example, polling for scoring updates on a basketball game at a frequency of 1000 times a second may provide little benefit where the frequency of scoring is tied closely to a 24-second shot clock.

445. 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 . . .":

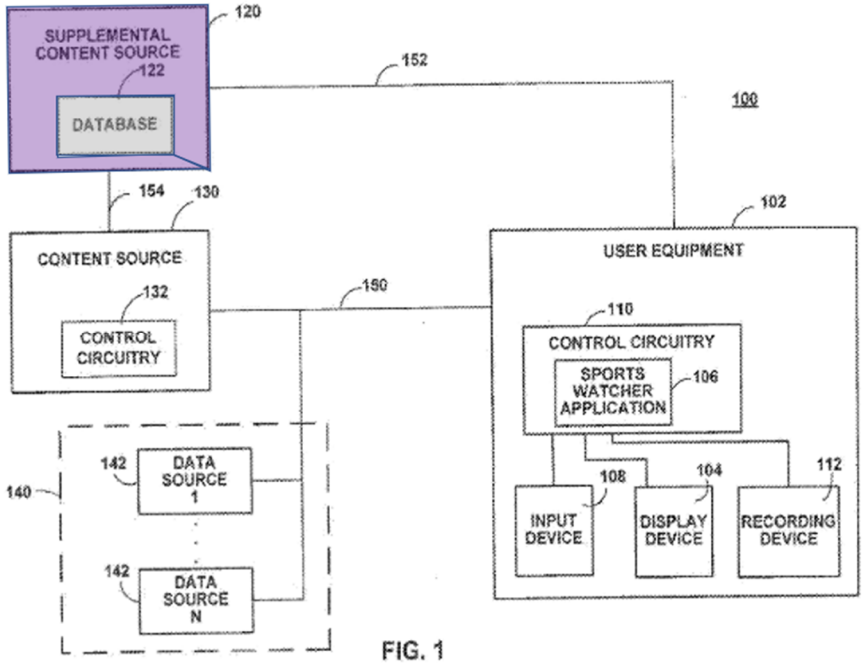


FIG. 1

Ex.1016 ¶42, Fig. 1. To the extent Ellis does not expressly disclose or render obvious this element, it is my opinion that Spivey's live data server device ("LDSD"), as shown in Figure 4, contains the same basic components as the '687's control server:

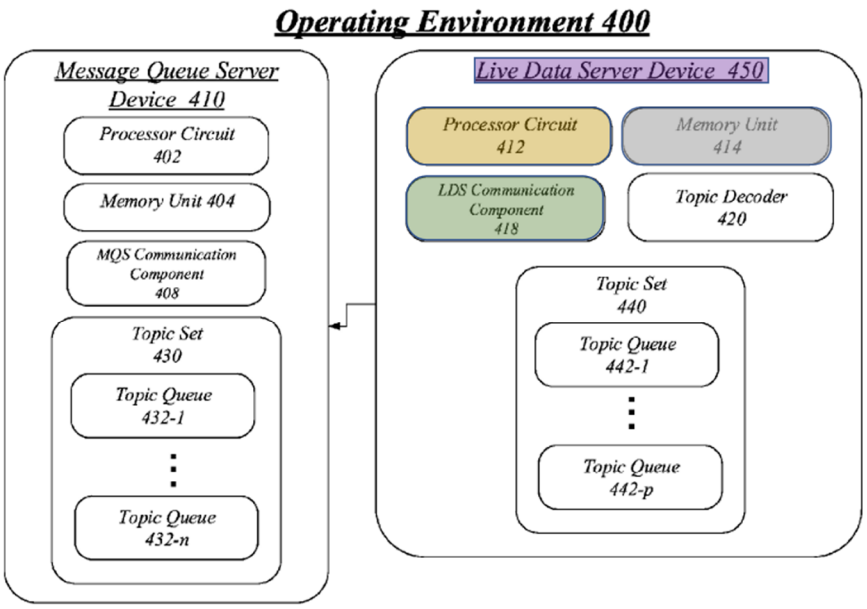


FIG. 4

Ex.1017, ¶¶54-55, Fig. 4 (annotated).

446. The LDSD is a generic server comprising one or more processors, memory devices, and communication interfaces. *Id.* Moreover, it is my opinion that the LDSD performs the same functions as the '687's control server. The LDSD contains an LDS communication component that allows LDSD "to receive live data and transmit the data payloads to MQS device 410." Ex.1017 ¶55. The live data received by the LDSD includes "sports scores and breaking news" and when related to baseball can include "a score, a play, an error, an injury, a player-at-bat, and so forth." *See id.* ¶¶27, 48, 66 ("A data payload may include … information about the live event in progress, such as a score, a name of the player that scored or made a play, an injury, a penalty, an error, a new record set, and so forth."). The LDSD retrieves the first event information germane to a first/second live sporting event

including the first/second score information for the first/second 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. Spivey also discloses a "Memory Unit 414" for storing the event information. Thus, it is my opinion that a POSA would have understood that Ellis and Spivey both disclose this element. To the extent that Ellis's SCS did not meet this claim element, I believe that a POSA would have motivated to substitute Ellis's SCS with Spivey's LDSD and MQSD for several reasons, including to decrease latency, and a POSA would have had a reasonable expectation of success because this would have involved nothing more than a simple substitution/combination of well-known, compatible elements to achieve a predictable result.. §§IX.A *supra.*

vi. **Element 1.6: "C) at least one socket server communicatively coupled to the control server to:"**

447. In my opinion, Ellis in view of Spivey discloses a socket server communicatively coupled to the control server (which, as discussed above, is disclosed by Ellis's SCS and Spivey's LDSD). The '687 describes the socket server as a generic server. *See* Ex.1003, 55:15-25. The purpose of the socket server is to "facilitate communication" of "synchronized event information retrieved by the control server" associated with an event between the control server and the VCDs.

Id., 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.,* 20:15-12. The socket server then receives event information transferred from the control server and shares the event information in "event messages" via the event information channel "in a synchronized manner" to all viewers "following the particular event." *Id.*

448. Ellis discloses providing event information over a separate connection to user equipment. Ex.1016 ¶¶39-41, Fig. 1. Ellis also discloses a system where multiple servers may provide the event information. Ex.1016 ¶47 (noting the SCS's functionality "may be … provided by multiple systems at multiple facilities").

449. In my opinion, a POSA would have looked to Spivey to add additional backend architecture to reduce latency in the real-time data delivery system. Like the '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, it is my opinion that a POSA would have understood that Spivey's MQSD carries out the same purpose as the '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." *Id.* ¶¶36; *see id.* ¶¶45 ("[MQS] device 310 may be a representative example of message queue server 120"), ¶54 (describing that 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:"

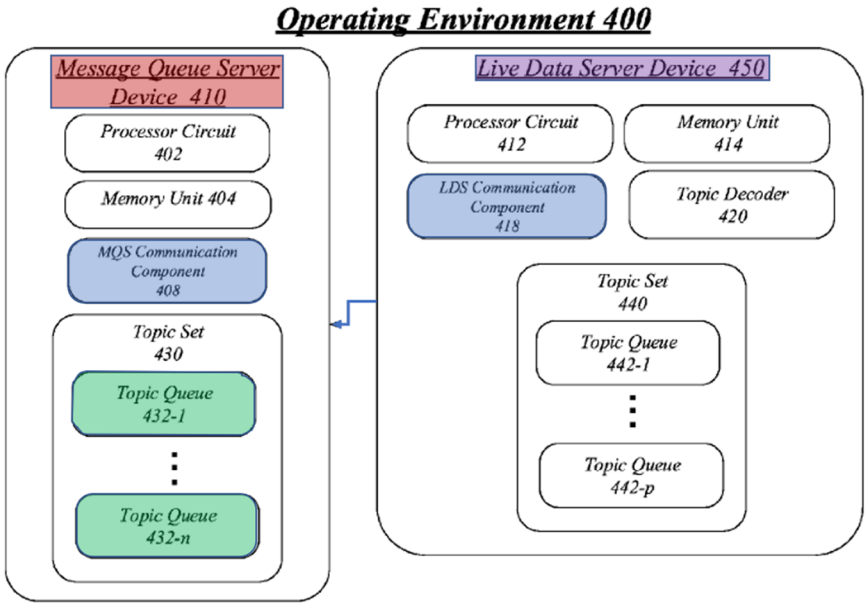


FIG. 4

Ex.1017, ¶¶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 (disclosing that communication component 308 transmits and receives data), ¶54 (disclosing that communication component 408 is analogous to 308), ¶55 ("LDS communication component 418 … allows [LDSD] to … transmit the data payloads to [MQSD]."). Then, 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. After sorting the data into topic queues, the MQSD shares the information "substantially simultaneously" to "the subscribed connected client

devices." *See id.* ¶96. Thus, it is my opinion that a POSA would have understood Spivey's MQSD discloses the '687's socket server being communicatively coupled to a control server.

450. Alternatively, Spivey teaches that latency associated with the broadcast of event data, such as sports scores, to clients can be reduced by "using topic cues and open duplex communication connections." *Id.* ¶¶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."). As an example of such "open duplex communication connections" Spivey discloses that communications between a server and a client 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"); *see also*, *Id.* ¶¶ 43, 61, 74, and 92; claim 8. Thus, it would also have been obvious to a POSA in view of these teachings of Spivey 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. In my opinion, a POSA would have recognized that doing so would further the express goals of Spivey with respect to reduction of latency in such transmissions. In my opinion, a POSA would have had a reasonable expectation of success implementing this web socket functionality in the MQSD because this would

have involved nothing more than a combination of well-known, compatible elements to achieve a predictable result.

vii. **Element 1.7: "receive from the control server at least the first event information; and"**

451. In my opinion, Ellis in view of Spivey discloses this element. The LDSD (the control server) "receive[s] live data and transmit[s] the data payloads to [MQSD (the socket server)]." Ex.1017 ¶¶55. Data payloads may include "a score, a player name, a play, and so forth …."

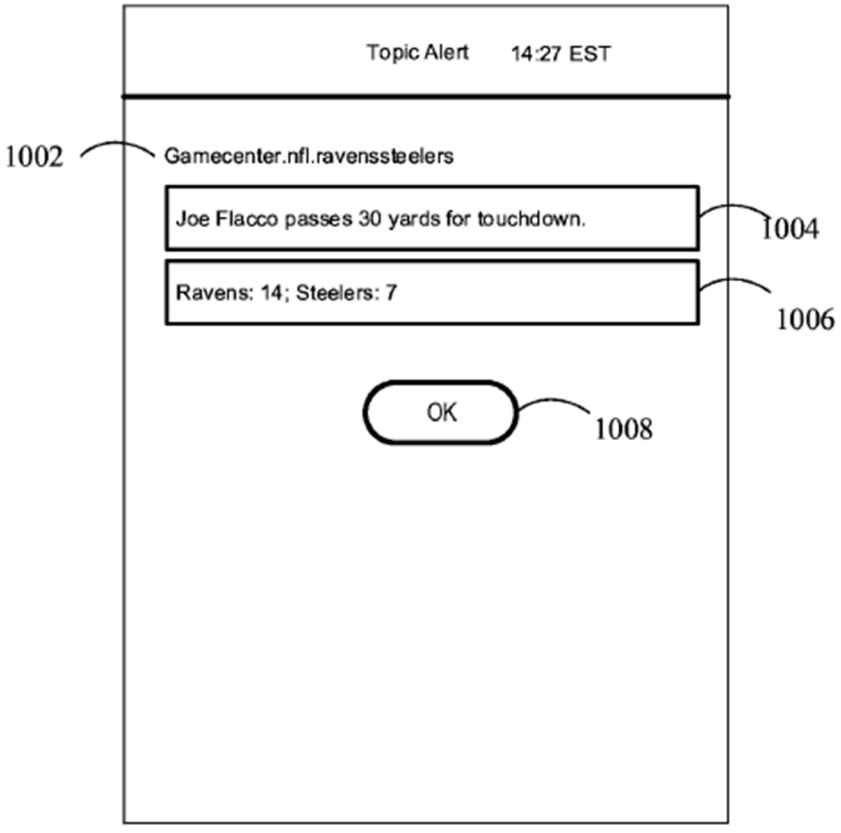


FIG. 10

Id. ¶85, 117, Figs. 9-10. Thus, it is my opinion that a POSA would have understood that Spivey discloses a socket server receiving from the control server at least the first score information. See §§IX.A.2, X.B.1.vi; *supra*.

viii. **Element 1.8: "transmit the first event information to at least the first viewer client device of the first plurality of viewer client devices 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 viewer client device of the first plurality of viewer devices, wherein the at least one first event socket corresponds to the first event information germane to the first live sporting event."**

452. In my opinion, Ellis in view of Spivey discloses this element. This element requires (1) a first socket on the socket server that is specifically dedicated to serving information about the first live event, (2) a first viewer client device of a group of VCDs that are watching a live stream of the first live event, and (3) transmitting event data about the first live event from the first socket to the VCD using an internet connection. *See* Ex.1003, Claim 1. These features are shown in Figure 2:

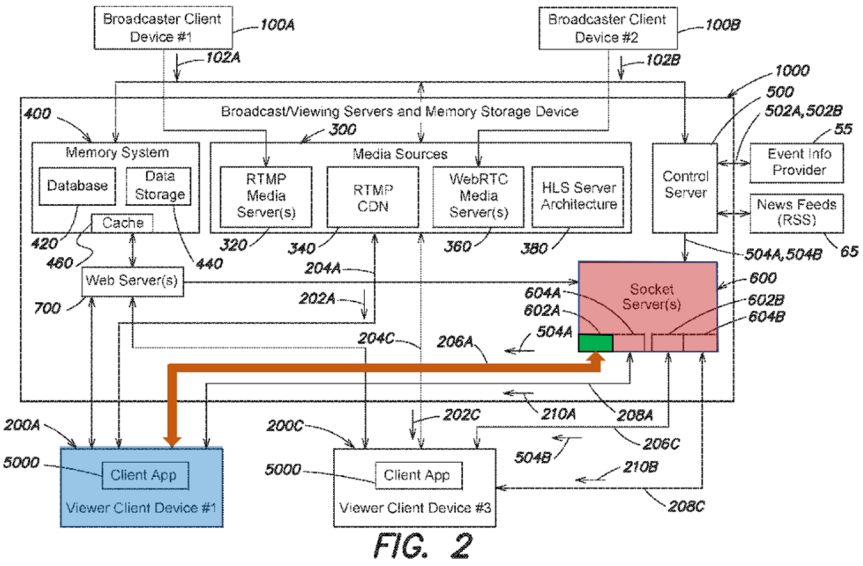


FIG. 2

Id., Fig. 2.

453. Initially, I would note that Spivey often discusses a connection between the client device and the system using a live data access server ("LDAS"), however, Spivey also discloses and teaches that the message queue server device may be "accessible to clients over the network directly." Ex.1017 ¶¶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. Also, the system does not require an LDAS, only that it "may include a[n] [LDAS]." *Id.* ¶38. Thus, it is my opinion that a POSA would have understood Spivey to disclose and teach a direct connection between the client devices and MQSD. In my opinion, a POSA would have used this implementation because reducing the amount

of server transfers would reduce latency, which is Spivey's main purpose. Furthermore, it was a well-known alternative to a POSA to implement the functionality of multiple servers as a single server. Ex.1016, ¶47.

454. In my opinion, a POSA would have understood that 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. *Id.* It is my opinion that 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. *See* Ex.1003 at 20:55-21:4, 21:63-22:16; Ex.1029, pp. 14-20; Ex1030; §IV.C.3., *supra*. In the same way, 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 then 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.

455. 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. *See 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:

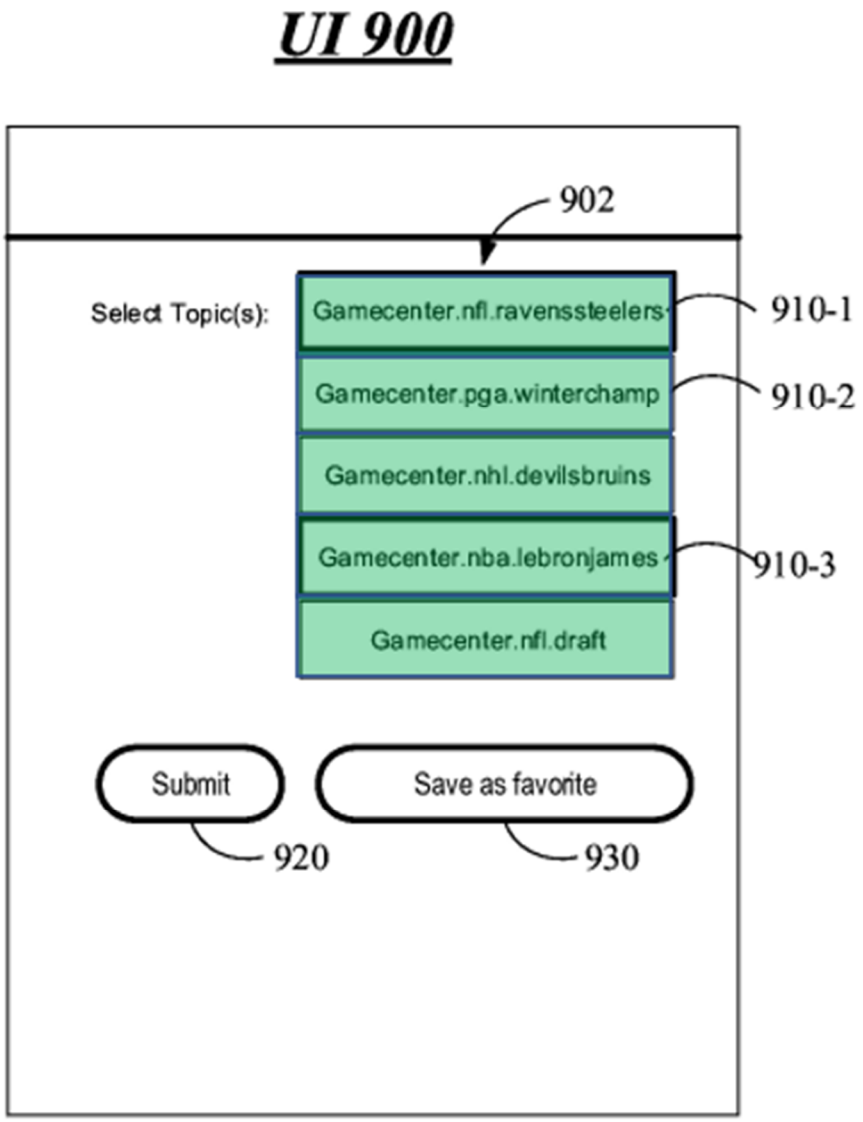


FIG. 9

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, it is my opinion that a POSA would have understood that Spivey's topic queues disclose the

'687's first event sockets corresponding to the first event information germane to the first live sporting event.

456. Alternatively, it would also have been obvious to a POSA in view of these teachings of Spivey 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 in §IX.A.

457. In my opinion, Ellis further discloses a first VCD of the first plurality of VCDs. *See* §X.B.1.a.i. Both Ellis and Spivey further disclose transmitting the first score 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. In Ellis, SCS may be configured "to provide real-time sports scores and game summary information, … to user equipment 102." Ex.1016 ¶41. Thus, it is my opinion that Ellis and Spivey disclose the above element.

458. In my opinion, Ellis in view of Spivey additionally 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 Internet." *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, it is my opinion that a POSA would have understood Ellis in view of Spivey to disclose transmitting first event information via a separate internet connection.

459. Further, it is my opinion that this information is provided via a connection between a topic queue and the first user equipment device. Users can access the MQSD using a URL. *See* Ex.1017 ¶¶81-82. Then, the user can access the specific topic queue she wishes to receive information from by using a text string, *e.g.*, a topic label or Event ID, associated with a topic queue. *Id.* ¶179, Fig. 9 ("Gamecenter.nfl.ravenssteelers"). Thus, it is my opinion that 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. Further, I believe 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.

2. '687 Ground 2: Claim 10 is Obvious Over Ellis in View of Spivey and Kellicker

460. For the reasons stated in §IX.B, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, and Kellicker to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 10

i. **Elements 10.0-10.2: "The system of claim 1, wherein: the plurality of media sources comprises: at least one of: at least one real-time messaging protocol (RTMP) media server to receive the first live stream of digital content; and at least one web real time communication (WebRTC) media server to receive the first live stream of digital content; and"**

461. In my opinion, Ellis in view of Spivey and Kellicker teaches these elements.

462. As explained in §IX.B above, Kellicker discloses a server (120/1250) that receives incoming live media streams (e.g., live incoming media stream (1220)), processes the incoming media streams and serves such media streams to playback devices in real-time. Ex.1049, ¶¶100-101, 106. Kellicker discloses that this server can be implemented through a plurality of such 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"), ¶24. Thus, Kellicker discloses a plurality of media sources in the form of real-time streaming media servers.

463. Kellicker also teaches that the server(s) may support various streaming protocols including, RTMP, and WebRTC. *Id*., ¶106. ("computing device 1250 may also support real time messaging protocol (***RTMP***) (and variants thereof), real-time streaming protocol (RTSP), real-time transport protocol (RTTP), web real-time communication (***WebRTC***), and MPEG-2 transport stream (MPEG-TS)"), ¶ 24 (disclosing transcoders/transmuxers for transforming media streams). Thus, Kellicker discloses media sources in the form of a real-time RTMP media server and/or a real-time WebRTC server.

464. Kellicker teaches providing its server(s) with the capability to "enable the computing device 1250 to process data in accordance with multiple coding technologies and protocols." *Id*., ¶103-104. In my opinion, it would have been obvious to a POSA to have implemented the content source(s) 130 of Ellis in the form of a real-time RTMP media server and/or a real-time WebRTC server. In my opinion, 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.

465. To the extent required by the claims, in my opinion it would have also been obvious to a POSA to have modified Ellis to implement the content source(s) 130 in the form of multiple servers, including a RTMP server ***and*** a WebRTC server. A POSA would have recognized that the use of multiple servers, as taught by

Kellicker, would have provided the system with even more flexibility, capacity and adaptability to handle multiple different types of incoming media and comply with multiple types of protocols. *Id*., ¶98.

ii. **Element 10.3: "an HTTP live streaming (HLS) caching server architecture communicatively coupled to at least one of the at least one RTMP media server and the at least one WebRTC media server; and"**

466. In my opinion, Ellis in view of Spivey and Kellicker teaches this element.

467. Kellicker teaches that the at least one RTMP media server and/or the at least one WebRTC media server are communicatively coupled to one or more other servers 1280. Ex.1049, ¶102. For example, 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., an edge server), etc. *Id*.; FIG. 12. Kellicker also explains that the data transmitted to the servers 1280 includes "media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof)." *Id*.

468. Kellicker further discloses that the media streams received by the one or more other servers 1280 "may be associated with the same encoding format and transmission protocol or may be associated with different encoding formats and

transmission protocols." *Id*. and ¶24 (disclosing transcoders/transmuxers for transforming media streams). Kellicker also specifically teaches providing the real-time streaming server(s) with the HLS streaming protocol ability. *Id*. at ¶¶105-106. In my opinion, it would have been obvious to a POSA to implement such HLS conversion capability in the form of the one or more other servers 1280, such as media/stream relay server 1281, for Ellis' content source(s). A POSA would have been motivated to provide HTTP live streaming (HLS) media streams because, as was well-known to a POSA, HLS "provides a reliable, cost-effective means of delivering continuous and long-form video over the Internet." Ex.1052, 2 (§1).

469. 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." *Id*. at ¶102. In my opinion, a POSA would have understood a CDN to be a caching architecture. Ex.1052, 2 (§1) ("[HLS] offers compatibility with large-scale HTTP caching infrastructure to support delivery to large audiences"); Ex.1055 Lepeska, 7:19-23 ("[a] content delivery network or content distribution network (CDN) is traditionally known as a system of computer containing copies of data placed at various nodes of a network, such that access to the cached data is improved by increasing bandwidth and reducing access latency"). To a POSA, the fundamental purpose of a CDN is to provide caching capability in order to improve responsiveness to client device requests for served data (in this case streamed media.

See also, Phillips (Ex.1050). Thus, in my opinion, a POSA would have understood that Kellicker teaches an HLS caching server architecture communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server.

iii. **Element 10.4: "the system establishes the first video Internet communication channel to provide the first copy of the first plurality of copies of the first live stream of digital content to the first viewer client device of the first plurality of viewer devices between the first viewer client device and the HLS caching server architecture."**

470. In my opinion, Ellis in view of Spivey and Kellicker teaches this element.

471. As can be seen below, Kellicker teaches communicating media streams between the HLS enabled other server(s) 1280 and client devices 1270:

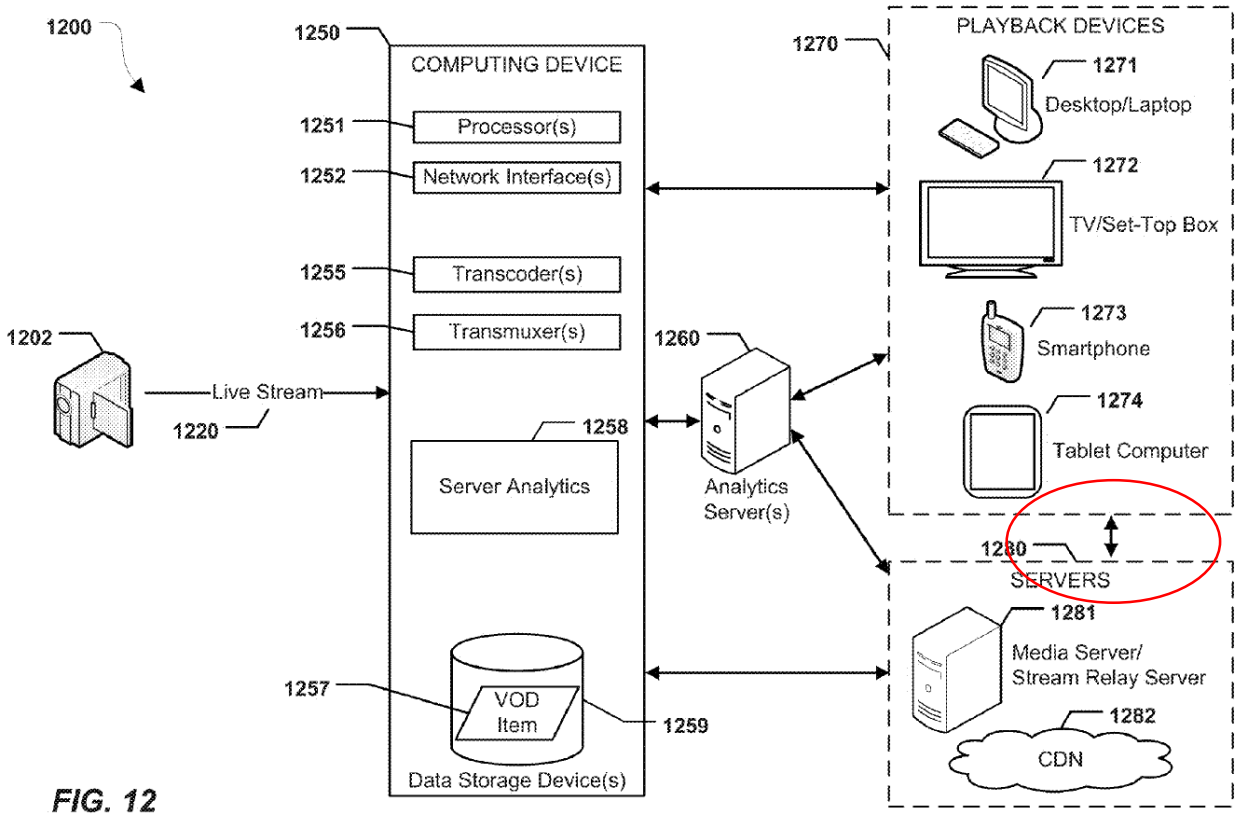


FIG. 12

Ex.1049, Fig. 12 (annotated) (illustrating an Internet communication channel between servers 1280 and playback/client devices 1270), ¶¶19, 98, 111, 102 ("data transmitted to the playback devices 1270 and to the servers 1280 includes media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof").

472. Likewise, Ellis teaches establishing a similar Internet video communication channel 160 between its contents source 130 and user equipment 102. See §X.B.1.a.i-iv.

3. '687 Ground 3: Claims 11 and 15-16 are Obvious Over Ellis in View of Spivey, Kellicker, and Phillips

473. For the reasons stated in §IX.C, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker and Phillips to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 11

i. **Elements 11.0-11.2: "The system of claim 10, wherein the HLS caching server architecture comprises: an HLS mother server communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server to receive, from the at least one of the at least one RTMP media server and the at least one WebRTC media server, a first HLS file suite representing the first live stream of digital content; anda plurality of HLS child servers communicatively coupled to the HLS mother server and an HLS content delivery network (HLS CDN),"**

474. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches these elements.

475. As explained above at §X.B.2., Kellicker teaches an HTTP live streaming (HLS) caching server architecture including a CDN communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server. It would have been obvious to a POSA to have provided this HLS caching server CDN 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." Ex.1050, ¶31. Consistent with the teachings of Phillips, a POSA

would have been motivated to provide this server architecture in order to provide "for high-performance streaming . . . to subscribers using one or more Internet-based infrastructures." *Id*., ¶30. For example, Phillips teaches that 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 implement Phillips' 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 . . .").

476. Kellicker as modified by Phillips establishes a communication path from a first 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, coupled to an HLS CDN architecture (1282), and from the CDN to a client/user device (1270):

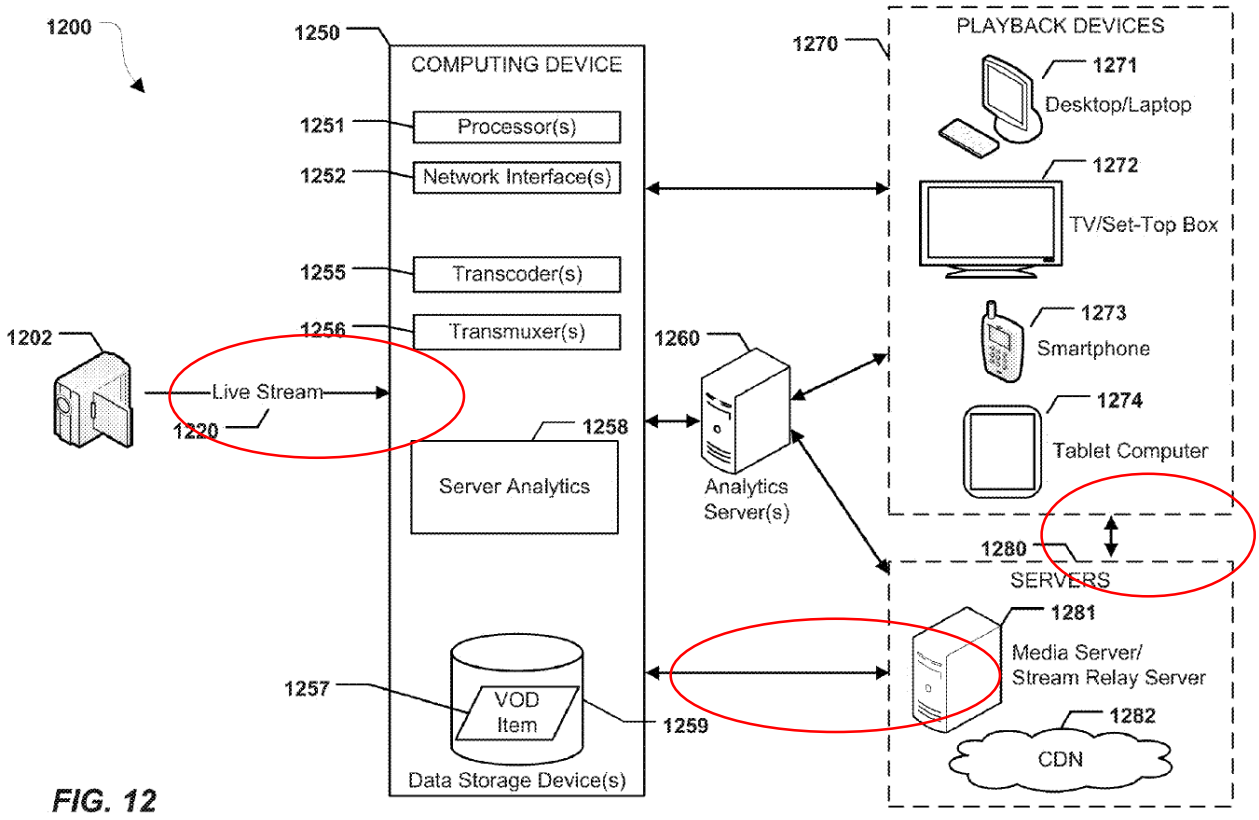


FIG. 12

Ex.1049, Fig.12 (annotated).

477. In particular, Kellicker discloses that the media server(s) 1250 is configured to transmit data to "a server of a content distribution network (CDN) 1282." *Id*., ¶102. Further, Kellicker discloses that, "[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.

478. In similar fashion, Phillips teaches that content that is 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 that 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 contents and/or nature of the files being handled by the system could be 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.1050, §§8.4-8.6. As can be seen, Phillips’ 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.

479. With respect to the "HLS file suite" this represents nothing more than conventional HLS file transfer protocol, as acknowledged in the ‘687. 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.1050, §§8.1-8.7. By delivering content from Kellicker’s server 1250 to a Phillips-type CDN, the HLS caching server architecture

comprises: an HLS mother server (e.g., national server, regional servers) which are communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server.

480. Hence, 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 live stream of digital content." A POSA would have understood that 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. As explained in the '687, an "HLS file suite" represents nothing more than the contents of a conventional HLS file transfer protocol. Ex.1003, 38:1–9. In my opinion, 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.1052, 27-29 (§ 8.1-8.7).

481. Alternatively, and to the extent that the nature or content of the data could be considered as limiting the claimed system, 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.1050, §§8.1-8.7. For example, as explained in Kellicker, this HLS caching server architecture receives a portion of the first HLS file suite (video stream) representing the first live stream of digital content. Ex.1049, ¶102 ("data

transmitted to the playback devices 1270 and to the servers 1280 includes media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof"). Further, Kellicker discloses that "first HTTP port(s) may be used to communicate a master rendition list and media chunks . . .," each of which is a portion of the HLS file suite. Ex.1049, ¶110; *see also* Ex.1003, 38:1–9.

482. Similarly, Phillips teaches that the first HLS file suite is received (media chunks and manifest/playlist). For example, Phillips teaches that content that is requested for delivery by the CDN may include a received "manifest" that is made available to a "content delivery node" (e.g., "mother" or "child" server) of the CDN server architecture. Ex.1050, 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"). And, to the extent that 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 (e.g., a "child" server). Ex.1050, Abstract, ¶31, Fig. 1. As another example, Phillips explains that the media portions of the HLS file suite are received by the CDN architecture. Ex.1050, FIG. 12 ("Pull Missing Segments for ABR Assets", which are media chunks of the HLS file suite), ¶9.

ii. **Element 11.3: "wherein: the system establishes the first video Internet communication channel to provide the first copy of the first plurality of copies of the first live stream of digital content to the first viewer client device of the first plurality of viewer devices between the first viewer client device and the HLS CDN; and"**

483. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches this element.

484. As explained above in §X.B.1.a.iv. 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. As further explained at §§XA.2.a.ii.-iii. and X.B.3.a.i., Ellis as modified by Spivey, Kellicker and Phillips establishes that the first video Internet communication channels are provided between the first viewer client device and the HLS CDN. *See also* Ex.1050, ¶33 ("Each local edge server is operable to serve a plurality of subscribers in a serving area for delivering content to associated end stations or client devices, as illustrated by an example client device or user equipment (UE) device 105 that may for consume or access streaming content delivered via CDN 100 in any type or number of access technologies including broadband access via wired and/or wireless (radio) communications."), ¶28 ("client devices may also access or consume content/services provided over… packet-switched wide area public network such as the Internet via suitable service provider access networks.").

iii. **Element 11.4: "the first copy of the first plurality of copies of the first live stream of digital content is based on the first HLS file suite."**

485. As explained at §X.B.3.a.i., a POSA would have understood that Ellis as modified by Spivey, Kellicker, and Phillips establishes that the first copy of the first live stream of digital content is based on the (first) HLS file suite as required by the HLS specification and the CDN distribution process.

b. Dependent Claim 15

i. **Element 15.0: "The system of claim 11, wherein 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 representing the first live stream of digital content."**

486. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches these elements.

487. In my opinion the claimed HLS mother/child server architecture is no different than the HLS CDN caching server architecture disclosed and taught by Phillips. Therefore, a POSA would have understood that the structure of Phillips' CDN hierarchy is certainly capable of executing the intended use or functionality recited in claim 15.

488. In my opinion, a POSA would have understood the recitation that 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 representing the first live stream of digital content" requires nothing more than the conventional functionality of a conventional caching CDN network.

489. For example, Ex. 1054 - Basile (EP2920708) discusses the purpose of the CDN, which is to avoid burdening the origin server with requests, by caching the content:

> In operation, end user device 120-121 request network content, such as content 135 associated with origin server 130. Instead of these requests being handled by the origin server 140, content node 110 of a content delivery network receives the content requests over ones of links 140-141 and processes the content requests for delivery of the content to the associated end user devices 120-121. Requested network content that is already stored in content node 110 can be provided quickly to the end user devices, while network content that is not already stored in content node 110 can be responsively requested by an associated content node 110 from origin server 130 for caching by content node 110. In this manner, content node 110 can act as an intermediary proxy node to provide local and fast access for end user devices 120-121 to network content of origin server 130 without burdening origin server 130 with many end user content requests.
> *Id*. ¶16.

490. Consistent with this, Phillips discloses a national server 102 of its CDN network that is an exemplary mother server. Ex.1050, ¶¶31, 32, 56, Fig. 12. In my opinion, a POSA would have understood that the national/mother server of Phillips can act as the primary server where the original/complete version of the served content can be stored, acting as a source for content distributed by the Content

Delivery Network (CDN) which can then distribute locally cached copies of that content from geographically distributed edge servers closer to users for faster access. In other words, a POSA would have understood that the national/mother server of Phillips can act as a backend "mother" server that holds the master copies of content that is served through the CDN network. Ex.1050, ¶61 ("If the missing audio, video, and/or subtitle asset segments are within the historical delivery trends/thresholds, such segments may be pulled from a regional or national server node or another edge server node (block 1072).").

491. Additionally, as explained in Phillips, portions of the HLS file suite can come from anywhere in the CDN. Ex.1050, ¶9 ("if any segment files referenced by the manifest are absent from the database…. pull from another content delivery node of the CDN only the representations of the absent segment files satisfying the one or more delivery rules and store the pulled representations in the database cache."), ¶8 ("After determining the compliant representations of 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").

492. Such a design avoids retrieving from the media servers any portions of the HLS file suite that exist anywhere in the CDN. Such operation is disclosed in Phillips. Ex.1050, ¶60 ("When all the loops have been performed for each adaptive streaming manifest in cache, for each child manifest in the master manifest, and for

each URL in each child manifest, all segment files should exist in the edge node's local cache memory. Upon verification that all segment files for all manifests exist in the edge node's local cache memory at block 1034, the process moves to block 1036 where the edge node reports to its parent node that defragmentation is complete."). Therefore, because all the segments exist somewhere in the CDN (e.g., at an edge node), no other HLS file suite representing the first live stream of digital content need be requested from the media server.

493. Based on at least the above, in my opinion, it would have been obvious to a POSA, in view of Phillips, to have used the national/mother server to request "only" the first HLS file suite associated with a 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 in a CDN. A POSA would have understood that this implementation allows data being served throughout the network to be replicated throughout and served by the CDN, which reduces load on the content creator server structure, and facilitates the defragmentation objective taught by Phillips.

c. Dependent Claim 16

i. **Element 16.0: "The system of claim 15, wherein the HLS mother server maintains a persistent connection to the at least one of the at least one RTMP media server and the at least one WebRTC media server, during at least a portion of a duration of the first live stream of digital**

content, to receive respective elements of the first HLS file suite."

494. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches these elements.

495. In my opinion, claim 16 is satisfied on the same basis as claim 15 above, because any connection between the HLS mother server and the media sources would "persist" over some unspecified duration of time.

496. In my opinion, based on the '687, a POSA would have understood that live media stream transmissions, including HLS, RTMP and WebRTC protocols, utilize a "persistent" connection between the source of the media and the server(s) distributing it to a user/client:

> 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.

497. But, according to the '687, "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, the connection of claim 16 does not refer to a connection between a server and a client device. Instead, claim 16 refers to the connection between the media source and the HLS CDN mother server.

498. Thus, as would be readily understood by a POSA, 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 utilizes a "persistent connection." In my opinion, a POSA would have understood that the "persistent connection" of claim 16 only need be persistent for "a portion of a duration of the first live stream of digital content."

499. Also, given the nature of the connection between the RTMP media server, the WebRTC media server, and the HLS mother server there would be very little reason to use a non-persistent connection, like an HTTP (request, response, close) connection, where a persistent connection would be more efficient and have lower overhead. For example, a POSA would have known that the creation and closing of connections requires a significant overhead via the allocation/deallocation of memory, use of additional system calls to the operating system, as well as the network overhead. While keeping the connection open prevents the reallocation of certain resources, this is acceptable given the nature of the connection between the CDN/mother server and the media servers as a streaming connection, that does not need to follow the HTTP protocol. In any event, for a POSA the choice between a “persistent” and “non-persistent” connection would have been nothing more than a typical tradeoff – implicating a design choice of increased overhead and potentially increased resource sharing vs. reduced overhead and reduced resource sharing.

500. Additionally, because the choice of connection is either persistent or non-persistent and both types of connections were well-known in the art, the choice of a persistent connect would have represented a mere design choice for a POSA.

501. Claim 16 also requires that the persistent connection is maintained during a portion of a duration of the first 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.

502. In my opinion, a POSA would have appreciated that the operation of claim 16 is consistent with live streaming itself. For example, if all elements of the first HLS file suite were to be received by the HLS mother server after the completion of the live stream, this would merely be more akin to playing a recorded video (VOD). Because Ellis in view of Kellicker and Phillips supports live streaming (as well as VOD) the HLS mother server (e.g., server 1280/1282, national server 102) would maintain a connection to the at least one of the at least one RTMP media server and the at least one WebRTC media server, during at least a portion of a duration of the first live stream of digital content, to receive respective elements of the first HLS file suite. See e.g., Ex.1049, ¶¶24, 28-30. Also, this condition will always be satisfied by a persistent connection that persists for the entirety of the live stream.

4. '687 Ground 4: Claims 12-14 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips, and Stoica

503. For the reasons stated in §IX.D, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker, Phillips and Stoica to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 12

i. Elements 12.0-12.1: "The system of claim 11, further comprising: at least one webserver to provide an applications programming interface (API) between the system and each of the first broadcaster client device, the second broadcaster client device, the first plurality of viewer client devices and the second plurality of viewer client devices; and;"

504. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element. To the extent claim 12 is interpreted as including: a first/second broadcaster client device, first live sporting event video-based commentary from the first broadcaster client device, and/or a broadcaster's live stream of digital content, etc. claim 12 is additionally met for the reasons stated in §§X.A.1-4.

505. Claim 12 specifies that a webserver provides an applications programming interface (API) between it and a number of devices of the system. However, the nature of the claimed API is unspecified. And, the claimed API has no further connection to claim 12 or any additional dependent claim. In my opinion, a POSA would have understood that a webserver may provide any of its APIs to any

system communicatively coupled to it, and, therefore, the webservers taught by Ellis in view of Spivey, Kellicker, and Phillips (as discussed above) disclose this claim element.

506. Also, Stoica expressly teaches at least one webserver that provides an API. For example:

> 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 (emphasis added).

507. Also, Stoica teaches a webserver in the form of a "dispatcher server" (1114) that may be 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.

508. Stoica's gateway 1122 and dispatcher server 1114 are each communicatively coupled with both client devices and the streaming content servers (e.g., CDN and origin servers). *Id.*, 9: 63–10:2; 11:17–32, 2:49–53; Fig. 11. As previously discussed, content requested by client devices, including first and second viewer content devices, is provided via a CDN, according to the system architecture taught by previous combinations of cited references, which included

communicatively coupled first and second pluralities of broadcast client devices providing streaming media content for distribution by the CDN. *See* §§VIII.A-C, X.B.1-3.

509. Stoica also teaches that the CMS is a webserver that provides an application 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 ("the CMS provides metadata associated with a video stream, including the information required to deliver the content.").

510. In its interactions between the CDN and client devices, Stoica teaches that the CMS/CDC can provide an augmented URI (i.e., VRL) specifying a manifest file to a client device:

> 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.").

511. It is clear that the CMS/CDC are communicatively coupled with both client devices and the streaming content servers (e.g., CDN and origin servers). *Id.*, Fig. 8, 2:49–53. As previously discussed, content requested by client devices, including first and second viewer content devices, is provided via a CDN, according to the system architecture taught by previous combinations of cited references, which included communicatively coupled first and second pluralities of broadcast client devices providing streaming media content for distribution by the CDN. *See* §§VIII.A-C, X.B.1-3.

ii. **Element 12.2: "a memory system communicatively coupled to the at least one webserver, the memory system including a database storing: first stream information for the first live stream of digital content including the first live sporting event video-based commentary from the first broadcaster client device, wherein the first stream information includes a stream identifier (StreamID) for the first live stream of digital content,"**

512. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

513. For example, Stoica teaches that the at least one of the webservers is communicatively coupled to a memory system including a database. In further detail, 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.

514. Stoica explains that this database stores a stream identifier (StreamID) (e.g., a "show" identifier), which is received for the first 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="NewsAtl0") OR (show="SundayMagazine").
> *Id*., 10: 31–49.

515. By storing a stream identifier (StreamID) for the first live stream of digital content the database stores first stream information for the first live stream of digital content including the first live sporting event video-based commentary from the first broadcaster client device because such data by definition includes the stream identifier (StreamID). *See* Claim 12 ("a memory system communicatively coupled to the at least one webserver, the memory system including a database storing: first stream information for the first live stream of digital content including the first live sporting event video-based commentary from the first broadcaster client device, wherein the first stream information includes a stream identifier (StreamID) for the

first live stream of digital content"). However, it is also clear that Stoica's database stores other information (e.g., viewer data) for the first live stream of digital content including the first live sporting event video-based commentary.

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

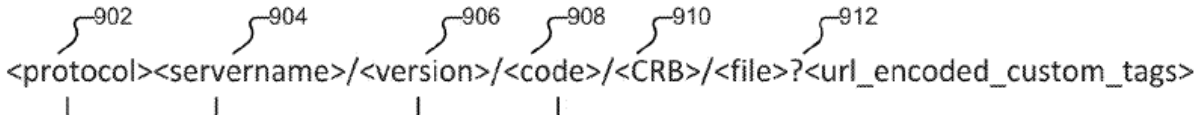
> Gateway 1122 also provides access controls and persistently stores the information regarding the API requests, data access and presentation policies, and filter descriptions. The information is maintained in a persistent database, such as mySQL or Oracle database.
> Ex.1051, 11:28-32.

517. Additionally, Stoica teaches that through the gateway 1112 API, data is available on a per filter basis. *Id*., 11:17-19. In my opinion, based on at least the above, a POSA would have understood that the stream identifier (StreamID) information is utilized for filtering, and ultimately routing content requests and retrieval. In my opinion, a POSA would have also understood that Stoica teaches that gateway 1122 is coupled to a database storing stream identifier (StreamID) information. The StreamID, which is an identifier for a particular source of content, *e.g.*, a "show" field in the case of Stoica.

iii. **Element 12.3: "wherein the at least one webserver: determines an address for the HLS CDN to establish the first video Internet communication channel to provide the first copy of the first plurality of copies to the first viewer client device, based at least in part on the StreamID for the broadcaster's live stream of digital content; and"**

518. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

519. 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 a 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."). Stoica FIG. 9 (excerpted below) depicts an exemplary VRL, which is transferred to client devices by a CMS webserver.



520. Next, the CDC determines the address for the HLS CDN, which is based on the VRL:

> 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 (emphasis added); FIG. 8 (illustrating transfer of video from CDN to client player beginning with a VRL).

521. Stoica further teaches that a number of optional parameters 912 may be embedded in the VRL, one of which is a "show" parameter (e.g., a StreamID). *Id*., 7:25-8:1-4 ("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 (discussed above) stored in the memory database of the RSP component (discussed above), which is communicatively coupled to the webserver portion of the dispatcher server.

iv. **Element 12.4: "transmits the address for the HLS CDN to the first viewer client device of the first plurality of viewer client devices to establish the first video Internet communication channel."**

522. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teach this element.

523. 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. *See e.g.,* Ex.1052, 24 (§6.3.1) ("How the client obtains the URI to the Playlist file is outside the scope of this document; it is presumed to have done so. The client obtains the Playlist file from the URI."):

> [I] nstead 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. . . . 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.

524. Once the client receives the playlist/manifest, the client may request a segment to establish the first video Internet communication channel to receive the

media stream. Ex.1052, 24 (§6.3.2) ("Every time a Media Playlist is loaded or reloaded from a Playlist URI, the client MUST determine the next Media Segment to load"), Ex.1051, FIG. 8 (illustrating transfer of VRL, manifest, and a video stream from a CDN to client player/device).

b. Dependent Claim 13

i. **Elements 13.0-13.1: "The system of claim 12, wherein: the first stream information for the first live stream of digital content further includes an event identifier (EventID) for the first live sporting event; and"**

525. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches these elements.

526. As explained above in §§X.B.1a.i-iv, Ellis in view of Spivey teach media source(s) that provide a first copy of a first live stream of digital content of a live sporting event to a viewer client device via a first video Internet communication channel. Ellis in view of Spivey also teaches 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. *See* §§X.B.1a.vi-viii; *See also*, Ex.1017, ¶92.

527. Spivey teaches 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. As can be seen, this topic label (EventID) "relates to" a specific sporting event or "show" ("ravenssteelers"), which is an example of a StreamID.

528. When further combined with the database system of Stoica, it would have been obvious for a POSA to store these EventIDs, in order to filter and store analytics information. Ex.1051, 9:46-10:54 ("Clients 1104-1112 each include a monitoring module that collects status information. As one example, performance information associated with video playback is collected, is other information about the client… the client is mapped to a dispatcher server…dispatcher server (1114) includes two conceptual modules. The first module implements a communication interface for receiving status information from clients… second task performed is to compute snapshots and on-line statistics over the telemetry data matching each filter").

529. In other words, storing stream information such as EventIDs amounts to nothing more than a design choice to transmit information already available to a client as taught by Spivey using mechanisms already provided by Stoica (e.g., dispatcher server and a database).

ii. **Element 13.2-13.3: "the at least one webserver further: determines a first socket address for the at least one first event socket of the at least one socket server based at least in part on the**

EventID; and transmits the first socket address to the first viewer client device of the first plurality of viewer client devices to establish the first event information Internet communication channel."

530. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

531. Spivey discloses that its MQSD and/or LDSD may be or include webservers. 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.

532. According to Spivey, "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").

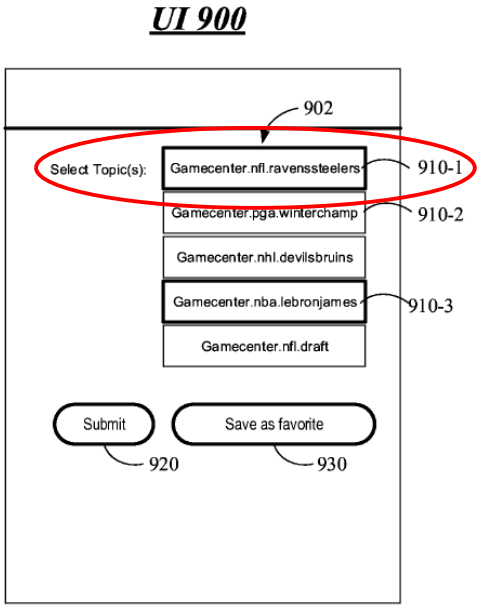


FIG. 9

533. In my opinion, 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." *See also*, §§X.B.1.a.vi-viii. 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.

534. As discussed in §§X.B.1.a.vi-viii above, it would have been obvious to a POSA in view of these teachings of Spivey for the MQSD server, which may be a webserver, to transmit the first socket address to the first viewer client device to establish a communication channel 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. Ex.1017 ¶¶53, ¶35, Fig.4.

535. In addition, as discussed above §§X.B.4.a.1.-iv, Stoica teaches an implementation of a webserver architecture as an intermediary between a client and a CDN. Stoica teaches that doing so facilitates communications and efficiency of the client's access to 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. In my opinion, based on the teachings of Stoica, it would have also been obvious to a POSA, to have transmitted the first socket address to the client to establish the first event information Internet communication channel via an intermediate web server such as Stoica's CMS or CDC.

c. Dependent Claim 14

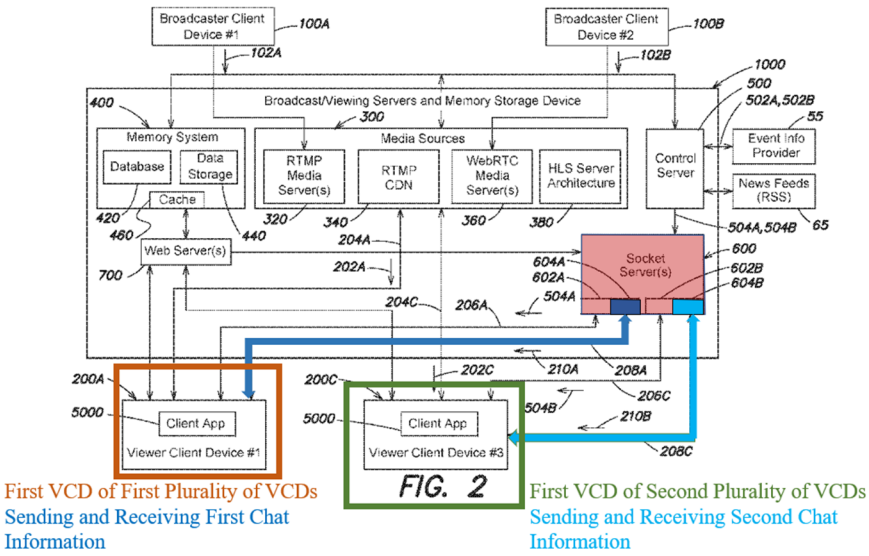
i. **Element 14.0-14.1: "The system of claim 13, wherein: in C), the at least one socket server further transmits and receives first chat information regarding the first live stream of digital content via at least one first chat/system event Internet communication channel between**

> **at least one first chat/system event socket of the at least one socket server and the first viewer client device of the first plurality of viewer client devices, wherein the at least one first chat/system event socket corresponds to the first chat information; and"**

536. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

537. As discussed above, 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. §§X.B.1.a.

538. The '687 explains that a chat/system event communication channel may be any generic internet communication channel. Ex.1003, 5:38-45. These elements only further require that the system "transmits and receives" first "chat information regarding" the first "live sporting event included in the first stream of digital content" over another separate, individual connection between a first "chat/system event socket" and a first VCD:



Ex.1003, 22:17-48, Claim 24, Fig. 2. In sum, it is my opinion that a POSA would have understood that these elements require that chat information is transmitted and received over a communication channel from another socket that is dedicated for chat/system information.

539. Ellis further 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. In my opinion, 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 live stream of digital content. Ex.1016 ¶126. See IX.D. *supra*.

540. 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.

541. In my opinion, as explained in §§X.B.1a.vi-viii, it would been obvious to a POSA to have established these communication channels/connections with web socket connections between the client and a web socket server.

542. 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. 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.

543. 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 would have been separate from the chat information according to the routing methodology applied by Spivey's socket server. *Id*.

544. In my opinion, based on the considerations above (hierarchy and exact matching) it would have also been obvious to a POSA in view of Spivey to provide a first chat/system event information socket (*e.g.*, "gamecenter.nfl.ravenssteelers.chat") corresponding to the first chat information regarding the first live sporting event video-based commentary.

ii. **Elements 14.2-14.3: "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 based at least in part on the StreamID; and transmits the second socket address to the first viewer client device of the first plurality of viewer client devices to establish the first chat/system event Internet communication channel."**

545. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Stoica teaches this element.

546. As discussed above, Spivey teaches that event information such as statistics would have been kept separate from chat information (on a separate channel/socket). 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 element 14.1.

547. In my opinion, in 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 and facilitating the use of hierarchical organization. Ex.1017 ¶¶48, 51-52. As discussed at §X.B.4.a.ii-iii above, a StreamID as discussed in Stoica would have been a "show." 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").

548. Alternatively, in my opinion using a StreamID rather than an EventID would have been 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.

549. As discussed in §§X.B.1a.vi-viii above it would have been obvious to a POSA in view of these teachings of Spivey for the MQSD server, which may be a webserver, to transmit the first socket address to the first viewer client device to establish a communication channel 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. Ex.1017 ¶¶53, ¶35, Fig.4.

550. In addition, as discussed above §X.B.4.a, Stoica teaches an implementation of a webserver architecture as an intermediary between a client and a CDN. Stoica teaches that doing so facilitates communications and efficiency of the client's access to 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. In my opinion, based on the teachings of Stoica, it would have also been obvious to

a POSA, to have transmitted the second socket address to the client to establish the first chat/system event Internet communication channel via an intermediate web server such as Stoica's CMS or CDC.

5. '687 Ground 5: Claims 17-18 are Obvious Over Ellis in View of Spivey, Kellicker, Phillips, and Pantos

551. For the reasons stated in §IX.E, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker, Phillips, and Pantos to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 17

i. **Elements 17.0-17.1: "The system of claim 1, wherein:";**

ii. **"the plurality of media sources comprises: at least one of: at least one real-time messaging protocol (RTMP) media server to receive the first live stream of digital content; and at least one web real time communication (WebRTC) media server to receive the first live stream of digital content; and";**

iii. **"an HTTP live streaming (HLS) caching server architecture communicatively coupled to at least one of the at least one RTMP media server and the at least one WebRTC media server;**

iv. **"the system establishes the first video Internet communication channel to provide the first copy of the first plurality of copies of the first live stream of digital content to the first viewer client device of the first plurality of viewer devices**

between the first viewer client device and the HLS caching server architecture;"

552. Elements 17.0-17.1 are substantially the same as elements 10.0–10.4. *See* §X.B.2.

v. **Element 17.2: "the HLS caching server architecture comprises: an HLS mother server communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server to receive, from the at least one of the at least one RTMP media server and the at least one WebRTC media server, a first HLS file suite representing the first live stream of digital content; and";**

vi. **"a plurality of HLS child servers communicatively coupled to the HLS mother server and an HLS content delivery network (HLS CDN);";**

vii. **"the system establishes the first video Internet communication channel to provide the first copy of the first plurality of copies of the first live stream of digital content to the first viewer client device of the first plurality of viewer devices between the first viewer client device and the HLS CDN;"; and**

viii. **"the first copy of the first plurality of copies of the first live stream of digital content is based on the first HLS file suite;"**

553. Elements 17.2 are substantially the same as elements 11.1– 11.4. See §X.B.3.a.

ix. **Element 17.3: "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 representing the first live stream of digital content;"

554. Element 17.3 is substantially the same as element 15.0. See §X.B.3.b

x. **Element 17.4: "the HLS mother server maintains a persistent connection to the at least one of the at least one RTMP media server and the at least one WebRTC media server, during at least a portion of a duration of the first live stream of digital content, to receive respective elements of the first HLS file suite;"**

555. Element 17.4 is substantially the same as element 16.0. See §X.B.3.c.

xi. **Element 17.5: "the first HLS file suite provided by the at least one of the at least one RTMP media server and the at least one WebRTC media server to the HLS mother server comprises: a first HLS playlist specifying a first plurality of different resolution versions of the first live stream of digital content, the first HLS playlist further including a corresponding first plurality of chunklist pointers to first respective chunklists for the first plurality of different resolution versions;";**

xii. **"the first respective chunklists for the first plurality of different resolution versions, wherein each chunklist of the first respective chunklists includes a first plurality of file segment pointers to first respective file segments for each different resolution version of the first plurality of different resolution versions; and"**

xiii. **"the first respective file segments for each different resolution version of the first plurality of different resolution versions;"**

556. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches these elements.

557. In my opinion, the above elements (17.5) refer to nothing more than claiming the contents of a master playlist configured in accordance with the known conventional HLS streaming protocol. To support the HLS format, the master playlists, media/video playlists (chunklists), and corresponding chunks ***must*** conform to the HLS Specification, otherwise the protocol is not HLS. Ex.1052, 4 (§4) ("Playlists that violate these rules are invalid; clients MUST fail to parse them.")

558. For example, Kellicker describes the use of "a master rendition list" or "manifest" (*i.e.*, a portion of 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"). As disclosed by Kellicker, this master rendition list may specify a first plurality of different versions (i.e., renditions) of the first live stream of digital content. Ex.1049, ¶27. Each rendition has a rendition chunklist and chunks (segments or fragments). *Id*., ¶¶4, 27. Kellicker further teaches that this protocol applies to both live and VOD streams. *Id*.

(A) Playlist specifying different resolution versions

559. Kellicker further teaches a plurality of renditions that are a plurality of different versions of the same media stream, *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.").

560. Similarly, Pantos recommends including resolution indications in master playlists for streaming video. Ex.1052, 15-17 (§4.3.4.2.) In my opinion, consistent with the HLS specification, it would have been obvious to a POSA to have included resolution variants in the HLS file suite in order 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.

(B) Chunklist pointers to respective chunklists

561. As seen from the exemplary master playlist taught by Pantos and reproduced below, a master HLS playlist includes a corresponding first plurality of chunklist pointers to first respective chunklists. In the example master playlist, 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").

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

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, 28 (§ 8.4).

(C) File segment pointers to respective file segments

563. As discussed 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., 28 (§ 8.2).

564. 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). *See also*, *Id*., 3 (§2) ("A Media Playlist contains a list of Media Segments, which when played sequentially will play the multimedia presentation."), 3 (§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.").

565. For other renditions/resolutions specified in the master playlist (for each different version of the first plurality of different versions), there would be a

corresponding set of file segment pointers and file segments in the corresponding media playlist. Ex.1052, 5 (§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").

566. As discussed above with respect to claims 10 and 11 §§X.B.2 and X.B.3.a, 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 server at the peak of the distribution hierarchy of an HLS CDN. The master playlist (master rendition list or portion of an 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 stream. In my opinion, for this reason alone, transfer of these constituent parts of the HLS file suite by RTMP media server and the at least one WebRTC media server to the HLS mother server would have been obvious to a POSA.

567. In addition, configuring the HLS file suite using a master rendition list or master playlist in the manner specified by the HLS Specification 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 consistent with both Kellicker and the HLS Specification.

568. As discussed above, 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 server at the peak of the distribution hierarchy of an HLS CDN. The master playlist (master rendition list or portion of an 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 stream. In my opinion, for this reason alone, transfer of these constituent parts of the HLS file suite by RTMP media server and the at least one WebRTC media server to the HLS mother server would have been obvious to a POSA.

xiv. **Element 17.6: "the HLS mother server caches the first HLS playlist, the first respective chunklists, and the first respective file segments and passes on to at least one HLS child server of the plurality of HLS child servers at least one first copy of the first HLS playlist, the first respective chunklists, and the first respective file segments;"**

569. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches this element.

570. A POSA would have understood that these elements are simply part of the CDN distribution process and architecture taught by Phillips. See §X.B.3.a. *See*

also 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)."). This is further confirmed by Ex. 1054 Basile. For example, ¶ 16 of Basile discusses the purpose of the CDN, which is to avoid burdening the origin server with requests, by caching the content:

> In operation, end user device 120-121 request network content, such as content 135 associated with origin server 130. Instead of these requests being handled by the origin server 140, content node 110 of a content delivery network receives the content requests over ones of links 140-141 and processes the content requests for delivery of the content to the associated end user devices 120-121. Requested network content that is already stored in content node 110 can be provided quickly to the end user devices, while network content that is not already stored in content node 110 can be responsively requested by an associated content node 110 from origin server 130 for caching by content node 110. In this manner, content node 110 can act as an intermediary proxy node to provide local and fast access for end user devices 120-121 to network content of origin server 130 without burdening origin server 130 with many end user content requests.
> Ex. 1054 ¶ 16.

571. Consistent with this, Phillips discloses a national server 102 of its CDN network that is an exemplary mother server. Ex.1050, ¶¶31, 32, 56, Fig. 12. In my opinion, a POSA would have understood that the national/mother server of Phillips can act as the primary server where the original/complete version of the served

content can be stored, acting as a source for content distributed by the Content Delivery Network (CDN) which can then distribute locally cached copies of that content from geographically distributed edge servers closer to users for faster access. In other words, a POSA would have understood that the national/mother server of Phillips can act as a backend "mother" server that holds the master copies of content that is served through the CDN network including child servers such as regional servers. Ex.1050, ¶61 ("If the missing audio, video, and/or subtitle asset segments are within the historical delivery trends/thresholds, such segments may be pulled from a regional or national server node or another edge server node (block 1072).").

xv. **Element 17.7: "the at least one HLS child server of the plurality of HLS child servers server caches the at least one first copy of the first HLS playlist, the first respective chunklists, and the first respective file segments and passes on to the HLS CDN at least one second copy of the first HLS playlist, the first respective chunklists, and the first respective file segments; and"**

572. In my opinion, this element is satisfied for the reasons elements 11.0-11.4, 15.0, 17.2 and 17.6 are satisfied. See §§X.B.3.a, X.B.3.b.i., X.B.5.a.v.-viii., xiv. above. Ex.1050, ¶¶8-9, 31, 59.

xvi. **Element 17.8: "the first copy of the first plurality of copies of the first live stream of digital content to the first viewer client device of the first plurality of viewer devices is based on the at least one second copy of the first HLS playlist, the first respective chunklists, and the first respective file segments."**

573. In my opinion, this element is satisfied for the reasons elements 11.0-11.4, 15.0, 17.2 and 17.6 are satisfied. See §§X.B.3.a, X.B.3.b.i., X.B.5.a.v.-viii., xiv. above. Ex.1050, ¶¶8-9, 31, 59.

b. Dependent Claim 18

i. **Elements 18.0-18.1: "The system of claim 17, wherein: the at least one of the at least one RTMP media server and the at least one WebRTC media server is configurable according to a plurality of HLS settings for generating the first HLS file suite, wherein the plurality of HLS settings include: a video segment duration for each file segment of the respective file segments; and"**

574. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches these elements.

575. Pantos teaches that segments contained in HLS playlists have a specified target duration. Ex.1052, 7 (§4.3.2.1) ("The EXTINF tag specifies the duration of a Media Segment. . . . This tag is REQUIRED for each Media Segment."). In my opinion, a POSA would have understood media servers to be "configurable" to set and include a video segment duration in the HLS file suite in order to comply with the HLS protocol. *See also* Ex.1056 Yang (U.S. Patent No. 9,973,785), 20:3-19 ("A segment size is a duration (e.g., length of time) for each MMSF, which may be determined by the ingest manager in order to divide the publishing stream into smaller MMSFs of the same duration. A playlist for a live

event may have a specified or target segment size, which may be a specific duration (e.g., 2 seconds) for each MMSF. A maximum segment size may be determined for each live event, where the maximum segment size is the maximum valid segment duration for the live event.").

ii. **Element 18.2: "an HLS buffer length including a first number of the first plurality of file segment pointers in each chunklist;"**

576. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches this element.

577. According to the HLS Specification, as taught in Pantos, the number of file segment pointers in a chunklist contained a media playlist may be limited. Ex.1052, 22 (§ 6.2.2) ("the server MAY limit the availability of Media Segments by removing Media Segments from the Playlist file (§ 6.2.1). If Media Segments are to be removed, the Playlist file MUST contain an EXT--MEDIA-SEQUENCE tag."). In my opinion, a POSA would have understood media servers to 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. *See also* Ex.1056, 19:56-20:2 ("A playlist length is a number of MMSFs that may be identified, by the media URI for each segment, in the playlist. A target playlist length may be a chosen number of media URIs to add to the playlist for that specific live event. The target playlist length may

act as a minimum number of media URIs, such that a playlist may be generated with more than the target number of media URIs, but will not generate a playlist with fewer media URIs than the target playlist length.")

iii. **Element 18.3: "the video segment duration for each file segment of the respective file segments is configured to be approximately two seconds or less; and"**

578. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches this element.

579. Pantos discloses, by way of example, a video segment duration of "less than 10 seconds" or "3.003 seconds." Ex.1052, 3 (§2). Two seconds is both less than 10 seconds and "approximately" 3.003 seconds. In my opinion, 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. *See also* Ex.1056, 3:41-54 ("each playlist contains a set number of MMSF URIs, where each MMSF URI provides the location of the MMSF in a data storage service. The MMSFs may each be of different or equal lengths. In some examples, MMSFs will be between 2-10 seconds in duration."), 20:3-19 ("A playlist for a live event may have a specified or target segment size, which may be a specific duration (e.g., 2 seconds) for each MMSF.").

iv. **Element 18.4: "the first number of the first plurality of file segment pointers in each**

chunklist for the HLS buffer length is one of four, five and six."

580. In my opinion, Ellis in view of Spivey, Kellicker, Phillips, and Pantos teaches this element.

581. According to the HLS specification, as taught in Pantos, "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, 22 (§6.2.2). Based on the HLS specification, 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 and just above the minimum requirements. In my opinion, 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. ¶. *See also* Ex.1056, 15:36-48 ("For example, the playlist may contain URIs of the four most-recent MMSFs of the live event stored at the data storage service"), 19:56-20:2 ("a target playlist length of four may correspond to four media URIs in the playlist. Each time the playlist is updated, a number n of the media URIs may be removed from the playlist and a new number z of media URIs may be added to the playlist, where the number of new media URIs added to the playlist must be four or more.").

6. '687 Ground 6: Claims 23 and 24 Are Obvious Over Ellis in View of Spivey

582. For the reasons stated in §IX.A, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker and Phillips to arrive at the claimed invention with a reasonable expectation of success.

a. Independent Claim 23

i. Element 23.0: "A method for providing, to a first viewer client device, a first copy of a first live stream of digital content relating to a first live sporting event, the method comprising:"

583. Essentially, Claim 23 is a method claim partially implementing the system of Claim 1. Claim 23 generally is broader requiring only a first instance rather than first and second instances (e.g., a first copy of a first live stream of digital content relating to a first live sporting event to a first viewer client device). *See* Ex.1003, Claims 1, 23. Regardless, Ellis in view of Spivey discloses a system where there are multiple live streams of digital content about multiple different live sporting events. Thus, a POSA would have understood that Ellis in view of Spivey discloses this element for the same reasons stated above in connection with claim element 1.0. §X.B.1.a.i., *supra*.

584. To the extent the preamble is limiting, it is my opinion that Ellis discloses this element. *See* Ex.1016, Abstract. Ellis discloses providing a first copy of live stream of digital content by providing "enhanced information access and

viewing functionality for sports-related programming," which includes "a sporting event, sports news, or sports commentary." Ex.1016 ¶¶8, 11, 38-40.

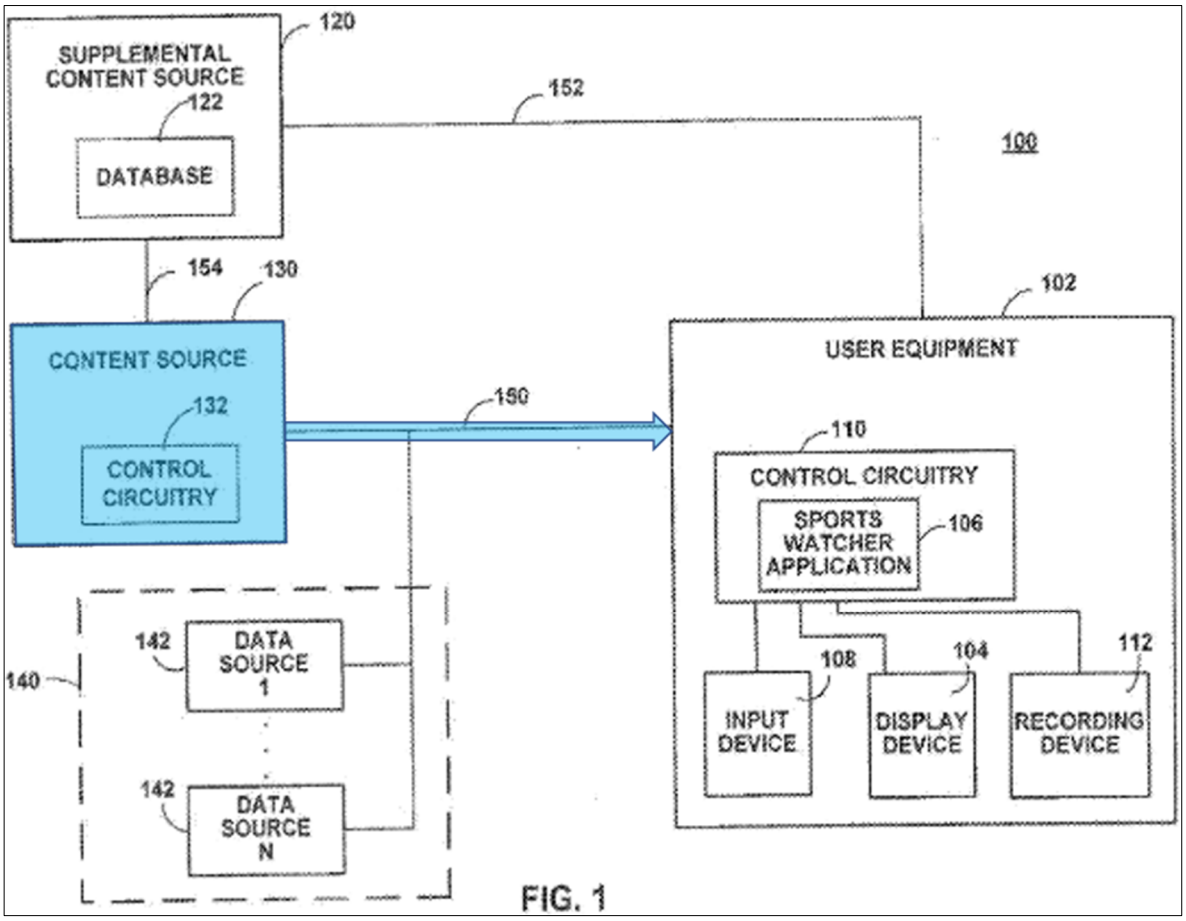
585. Ellis discloses providing the "media content" from those sporting events "to user equipment 102." *Id.* ¶¶38–40; Fig.1. Ellis defines "sporting event" as encompassing "online sports commentary." *Id.* ¶38. And, Ellis teaches providing sports commentary in a data feed or over communications channel to a user. *Id.*, ¶41. Further, media content includes broadcast programming "and any other content capable of being displayed by, presented to, … or interacted with, using user equipment 102." *Id.* ¶40.

ii. **Element 23.1: "A) providing the first copy to the first viewer client device via a first video communication channel between at least one media source and the first viewer client device;"**

586. In my opinion, Ellis and Spivey disclose this element. *See* Elements §§X.B.1.a.i.-iv. *supra*. These elements are substantially the same as the '687's element 1.3. Ex.1003, Claims 1, 23. The claim language is slightly altered, but the claims contain the same elements. Thus, a POSA would have understood that Ellis and Spivey disclose this element for the same reasons stated above in connection with at least claim elements 1.0-1.3.

587. Ellis discloses content sources, which are "any suitable source of media content" such as an "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 live stream of digital content and provide video content to user equipment via communications paths. *See id.* ¶¶39-41, 45, 47. Figure 1 shows a content source providing video content to user equipment over communication path 150:



Ex.1016 ¶¶39-41, Fig. 1 (annotated). Ellis also discloses multiple content sources, *i.e.*, a plurality. *Id.* ¶39. ("[T]here may be numerous instances of content source 130 …"); ¶74. As also apparent from Fig. 1 above, Ellis also discloses that users can be connected with a plurality of data sources 140. *Id*. Moreover, as explained below,

Ellis's content sources perform the same functions as the at least one media source as recited in 23.1.

588. As explained above, Ellis discloses providing individual copies to live streams to client devices. Ellis further discloses providing digital content or video content over a video communication channel. As an example of an "video communication channel," '687 describes a "video" channel between a media server and a client device to convey video-based commentary. Ex.1003, 5:45-49.

589. Ellis discloses providing video from a content source to user equipment, over a communication path. Ex.1016 ¶39, 74. The content source can be an "internet or network media/web server" and the user equipment "may include computer equipment." *Id.* ¶¶40, 48. The communication path can be "for example, a satellite path, a fiber-optic path, a cable path, or any other suitable wired or wireless path." *Id.* ¶45. 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 is provided with a copy from the content source over a first video communication channel that is unique to each user device. *Id.*

iii. **Element 23.2: "B) retrieving, via the Internet, first event information germane to the first live sporting event; and"**

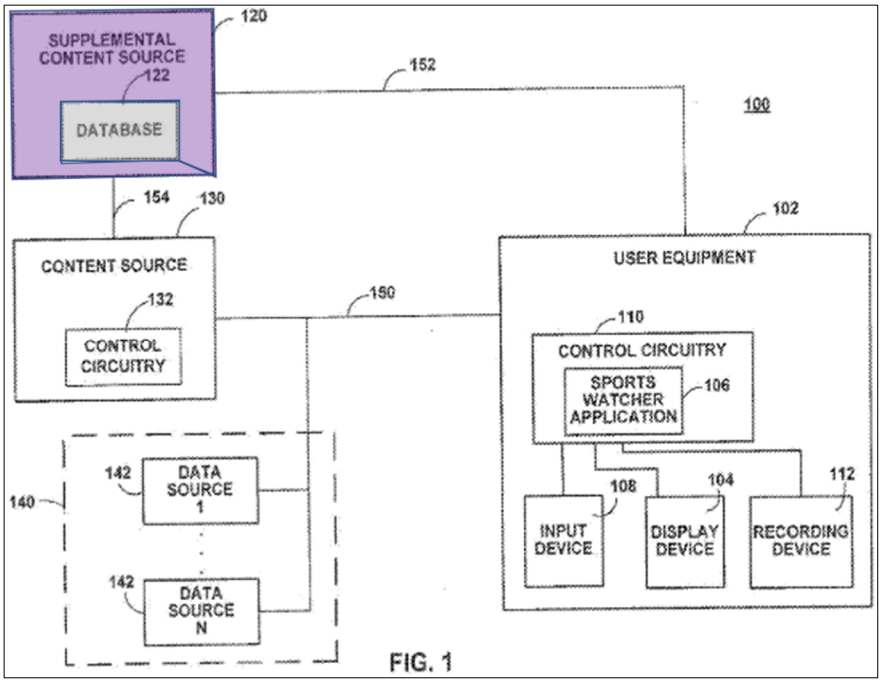
590. In my opinion, Ellis and Spivey both disclose this element. This element is substantially the same as the '687's element 1.5. Ellis and Spivey both disclose a multiple-sporting-event system that retrieves real-time data for all events. §X.B.1.a.v. *supra*. The claim language is slightly altered, but the claims contain the same elements. Thus, a POSA would have understood that Ellis in view of Spivey discloses this element for the same reasons stated above. §§X.B1.a.v. *supra.*

591. The '687 explains that the control server is a generic server comprising one or more processors, memory devices, and communication interfaces. Ex.1003, 55:15-25. The control server retrieves "event information" from an "event information provider." *Id.*, 21:47-50. The control server then transmits the event information to the socket server. *Id.*, 20:7-21, 21:55-62. 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.*, 21:48-54, 19:51-54. Further, the control server may "store at least some portion of retrieved event information." *Id.*, 44:43-45:3.

592. Ellis's SCS comprises one or more processors, memory devices, and communication interfaces. 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." Ex.1016

¶¶42, 142. The SCS retrieves, via the internet and from an event information provider, first event information germane to multiple live sporting events, including score information for each of the multiple live sporting events because the SCS may "provide real-time sports scores and game summary information, … spread and wagering data, and any other sports-related information to user equipment 102." *Id.* ¶41; see *id.* ¶¶72, 117. In my opinion, a POSA would have understood that this information is retrieved via the internet. *Id*. ¶¶37-38, 40, 142. Furthermore, a POSA would have understood that this was how real-time data was routinely transferred to users at the time of Ellis and Spivey—including score updates and wagering information. Ex.1016 ¶¶42, 142. Additionally, in my opinion, 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. Ex. 1024, 408-410.

593. 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).

594. To the extent Ellis does not expressly disclose or render obvious this element, it is my opinion that Spivey's live data server device ("LDSD"), as shown in Figure 4, contains the same basic components as the '687's control server. Ex.1017, Fig. 4; Ex.1003, 55:15-25.

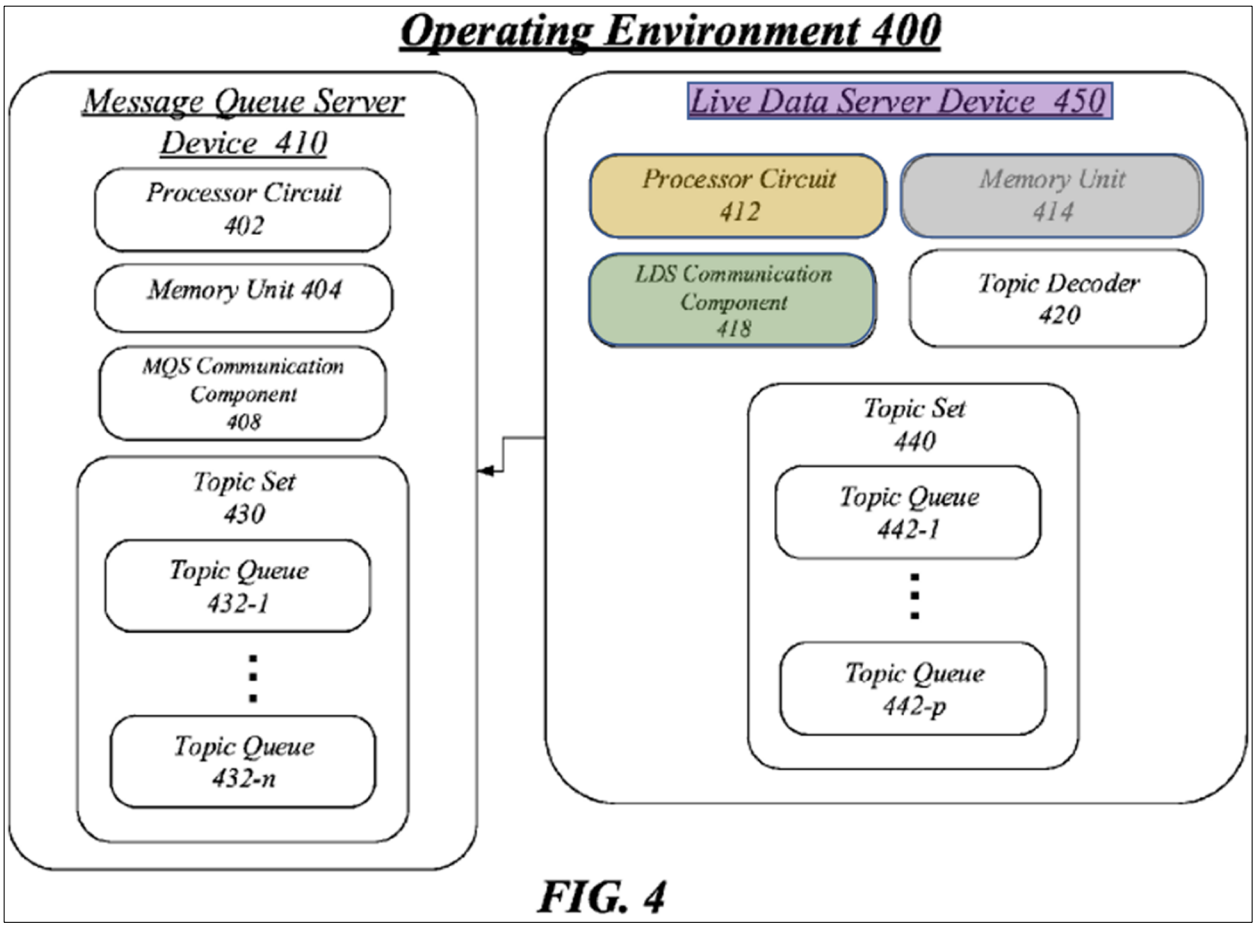


FIG. 4

Ex.1017, ¶¶54-55, Fig. 4 (annotated).

595. The LDSD is a generic server comprising one or more processors, memory devices, and communication interfaces. *Id.* In my opinion, the LDSD performs the same functions as the '687's control server. The LDSD contains an LDS communication component that allows LDSD "to receive live data and transmit the data payloads to MQS device 410." *Id.* ¶55. The live data received by the LDSD includes first event information related to at least first live event, e.g., "sports scores and breaking news" and when related to baseball can include "a score, a play, an error, an injury, a player-at-bat, and so forth." *See id.* ¶¶27, 48, 66 ("A data payload

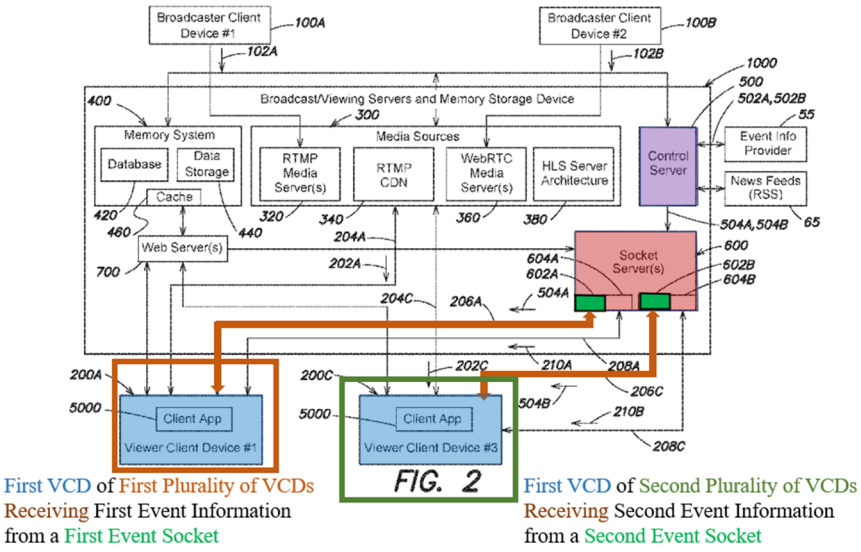
may include … information about the live event in progress, such as a score, a name of the player that scored or made a play, an injury, a penalty, an error, a new record set, and so forth."), Fig. 9. The LDSD retrieves the first event information germane to a first live sporting event including the 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. *Id*. ¶¶35, 37, 178. Thus, it is my opinion that a POSA would have understood that Ellis and Spivey both disclose this element. To the extent Ellis's SCS did not meet this claim element, it is my opinion that a POSA would have replaced Ellis's SCS with Spivey's LDSD and MQSD to decrease latency. *See* §§IX.A, X.B.1.v. *supra*.

iv. **Element 23.3: "C) transmitting the first event information to at least the first viewer client device via a first event information communication channel between at least one first event socket of at least one socket server and the first viewer client device, wherein the at least one first event socket corresponds to the first event information germane to the first live sporting event."**

596. In my opinion, Ellis in view of Spivey discloses these elements. These elements are substantially the same as the '687's elements 1.8 except that there is a different name for the communication channel. Ex.1003, Claims 1, 23. The different channel name alters nothing. §X.B.1.vi-viii *supra.* Ellis in view of Spivey

discloses a system of providing real-time data from multiple live games to multiple groups of users over separate (third/fourth) internet connection feeds. §§§X.B.1.vi-viii. The claim language is slightly altered, but the claims contain the same elements. Thus, a POSA would have understood that Ellis in view of Spivey discloses these elements for the same reasons as stated above. §§§X.B.1.vi-viii, §X.B.1.vi-viii, *supra*.

597. This element requires transmitting event information about the first live event from the first socket to the VCD. *See* Ex.1003, Claim 23. These features are shown in Figure 2:



Id., Fig. 2.

598. In my opinion, Ellis in view of Spivey discloses a socket server communicatively coupled to the control server (which, as discussed above, is disclosed by Ellis's SCS and Spivey's LDSD). The '687 describes the socket server

as a generic server. *See* Ex.1003, 55:15-25. The purpose of the socket server is to "facilitate communication" of "synchronized event information retrieved by the control server" associated with an event between the control server and the VCDs. *Id.*, 20:7-20:21. 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.,* 20:15-21. The socket server then receives event information transferred from the control server and shares the event information in "event messages" via the event information channel "in a synchronized manner" to all viewers "following the particular event." *Id.*

599. Ellis discloses providing event information over a separate connection to user equipment. Ex.1016 ¶¶39-41, Fig. 1. Ellis also discloses a system where multiple servers provide the event information. Ex.1016 ¶47 (noting the SCS's functionality "may be … provided by multiple systems at multiple facilities").

600. In my opinion, a POSA would have looked to Spivey to add additional backend architecture to reduce latency in the real-time data delivery system. Like the '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, it is my opinion that a POSA would have understood that Spivey's MQSD carries out the same purpose as the '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." *Id.* ¶¶36; *see id.* ¶¶45 ("[MQS] device 310 may be a representative example of message queue server 120"), ¶54 (describing that 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:"

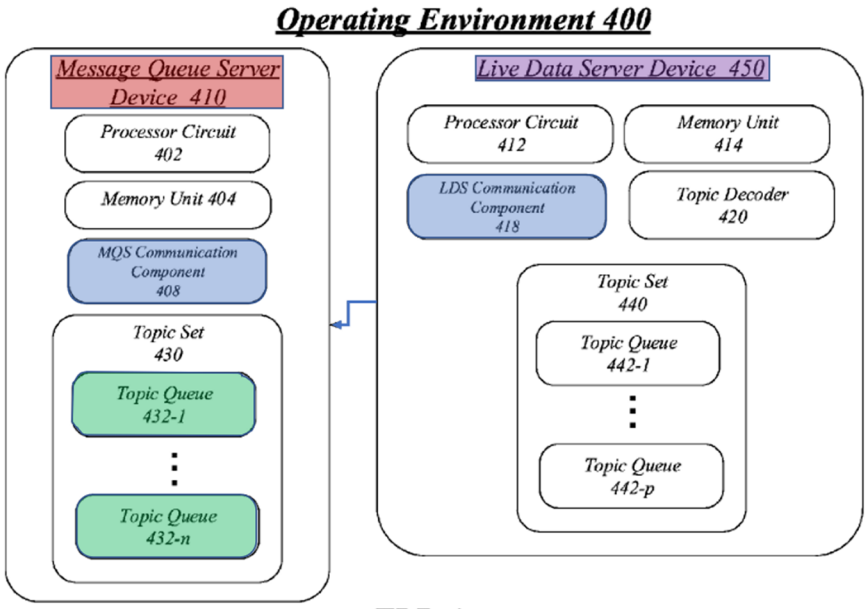


FIG. 4

Ex.1017, ¶¶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 (disclosing that communication component 308 transmits and receives data), ¶54 (disclosing that communication component 408 is analogous to 308), ¶55 ("LDS communication component 418 … allows [LDSD] to … transmit the data payloads to [MQSD]."). Then, 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. After sorting the data into topic queues, the MQSD shares the information "substantially simultaneously" to "the subscribed connected client devices." *See id.* ¶96. Thus, it is my opinion that a POSA would have understood Spivey's MQSD discloses the '687's socket server being communicatively coupled to a control server.

601. Alternatively, Spivey teaches that latency associated with the broadcast of event data, such as sports scores, to clients can be reduced by "using topic cues and open duplex communication connections." *Id.* ¶¶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."). As an example of such "open duplex communication connections" Spivey discloses that communications between a server and a client 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"); *see also, Id.* ¶¶ 43, 61, 74, and 92; claim 8. Thus, in my opinion, it would also have been obvious to a POSA in view of these teachings of Spivey 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. In my opinion, a POSA would have

recognized that doing so would further the express goals of Spivey with respect to reduction of latency in such transmissions. In my opinion, a POSA would have had a reasonable expectation of success implementing this web socket functionality in the MQSD because this would have involved nothing more than a combination of well-known, compatible elements to achieve a predictable result.

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

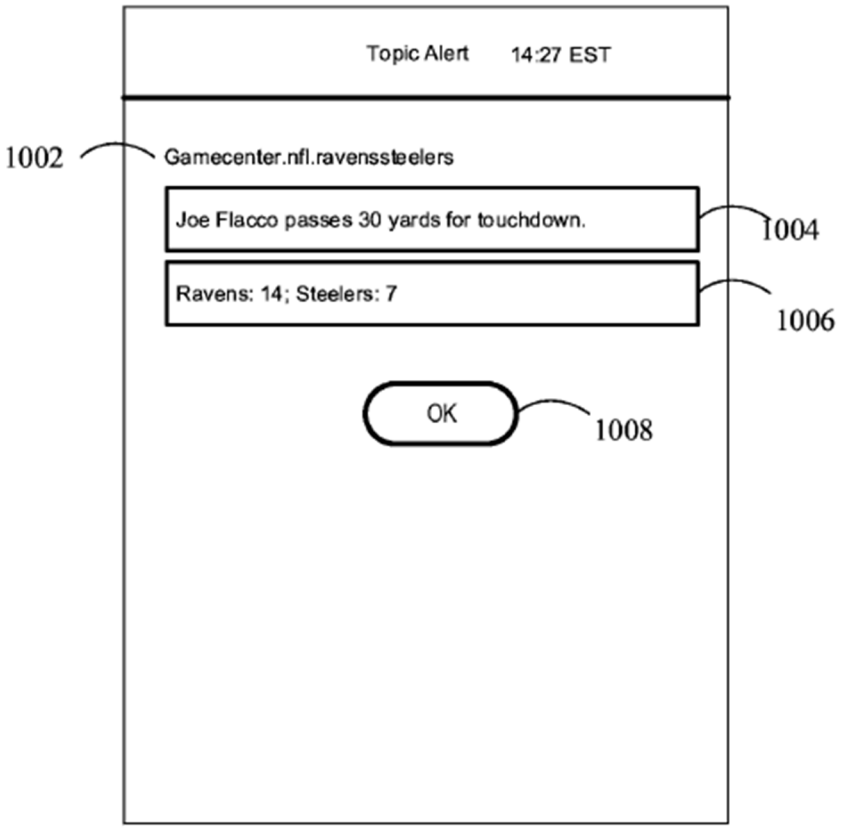


FIG. 10

Id. ¶85, 117, Figs. 9-10. Thus, it is my opinion that a POSA would have understood that Spivey discloses a socket server receiving from the control server at least the first score information.

(A) First event socket corresponds to the first event information germane to the first live sporting event

603. Spivey often discusses a connection between the client device and the system using a live data access server ("LDAS"), however, Spivey also discloses and teaches that the message queue server device may be "accessible to clients over the network directly." Ex.1017 ¶¶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. Also, the system does not require an LDAS, only that it "may include a[n] [LDAS]." *Id.* ¶38. Thus, it is my opinion that a POSA would have understood Spivey to disclose and teach a direct connection between the client devices and MQSD. In my opinion, a POSA would have used this implementation because reducing the amount of server transfers would reduce latency, which is Spivey's main purpose. Furthermore, it was a well-known alternative to a POSA to implement the functionality of multiple servers as a single server. Ex.1016, ¶¶4-7.

604. In my opinion, a POSA would have understood that 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. *Id.* It is my opinion that 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. *See* Ex.1003 at 20:55-21:4, 21:63-22:16; Ex.1029, pp. 14-20; Ex1030; §IV.C.3., *supra*.

605. In the same way, 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 then 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.

606. 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. *See 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:

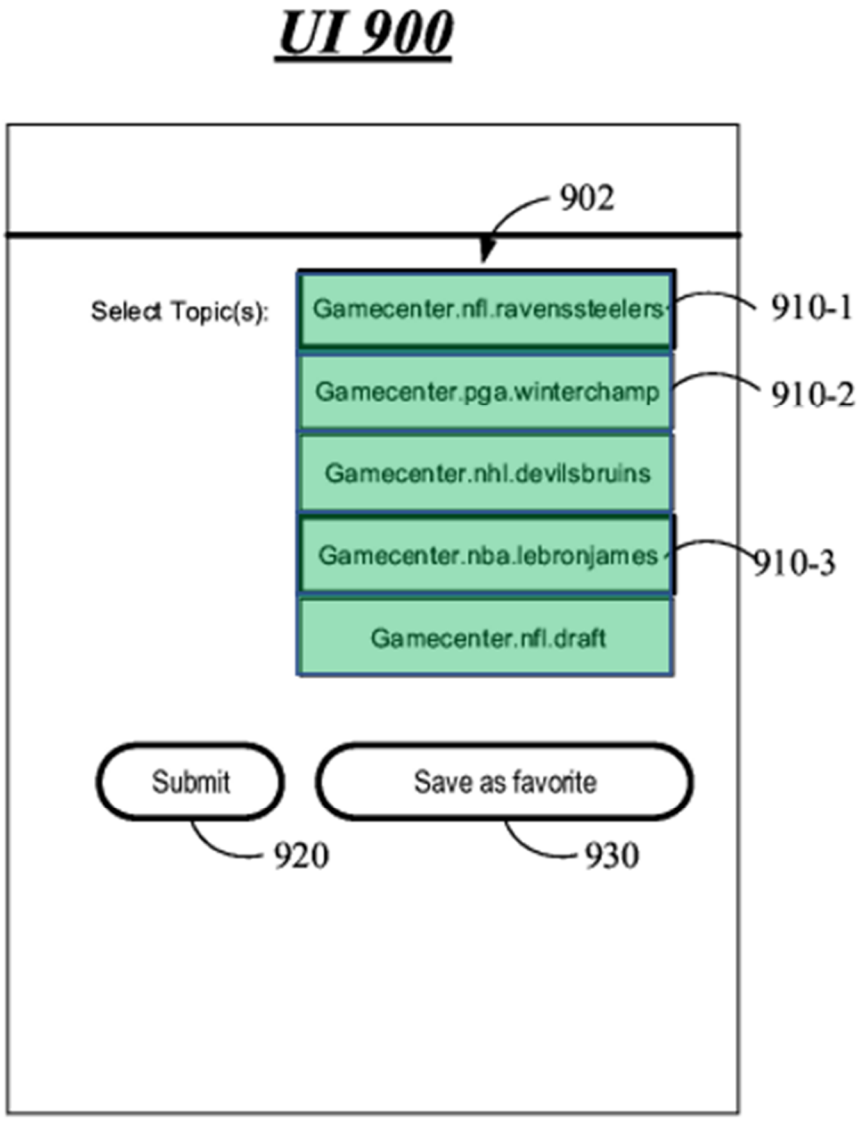


FIG. 9

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, it is my opinion that a POSA would have understood that Spivey's topic queues disclose the

'687's first event sockets corresponding to the first event information germane to the first live sporting event.

607. Alternatively, it would also have been obvious to a POSA in view of these teachings of Spivey 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 in §IX.A. and §X.B.1.a.vi.-viii.

(B) Transmitting the first event information to at least the first VCD

608. In my opinion, Ellis discloses a first VCD. *See* §§X.B.6.a.i-ii, X.B.1a.iv. *supra.* Both Ellis and Spivey further disclose transmitting the first event information to the first 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. In Ellis, SCS may be configured "to provide real-time sports scores and game summary information, … to user equipment 102." Ex.1016 ¶41. Thus, it is my opinion that Ellis and Spivey disclose the above element.

(C) Transmitting via a first event information 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

609. As previously noted, Ellis discloses at least a first VCD. *See* §§X.B.6.a.i-ii, X.B.1a.iv., *supra.* The signals are individual copies of the same live stream of digital content. Ex.1016 ¶¶40, 62, Fig. 2B. 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. *Id.* ¶45, 74. The communication path can be any "suitable wired or wireless path." *Id.* "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 live streams of digital content to a plurality of viewers from a plurality of broadcaster devices.

610. In my opinion, Ellis in view of Spivey discloses providing first event/score information over a separate 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 Internet." *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 than the video signal 150. Ex.1017 ¶39-41, Fig. 1. Thus, in my opinion a POSA would have understood Ellis in view of Spivey to disclose transmitting first event information via a separate internet connection, which includes the first event information communication channel.

611. Further, this information is provided via a connection between a topic queue and the first user equipment device of the first plurality of user devices. Users can access the MQSD using a URL. *See* Ex.1017 ¶¶81-82. Then, the user can access the specific topic queue she wishes to receive information from by using a text string, *e.g.*, a topic label or Event ID, associated with a topic queue. *Id.* ¶179, Fig. 9 ("Gamecenter.nfl.ravenssteelers."). Thus, in my opinion a POSA would have understood that Ellis in view of Spivey discloses distributing event information from a first event socket of the socket server to a first user device of a first plurality of user devices over an internet connection. Further, it is my opinion that 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.

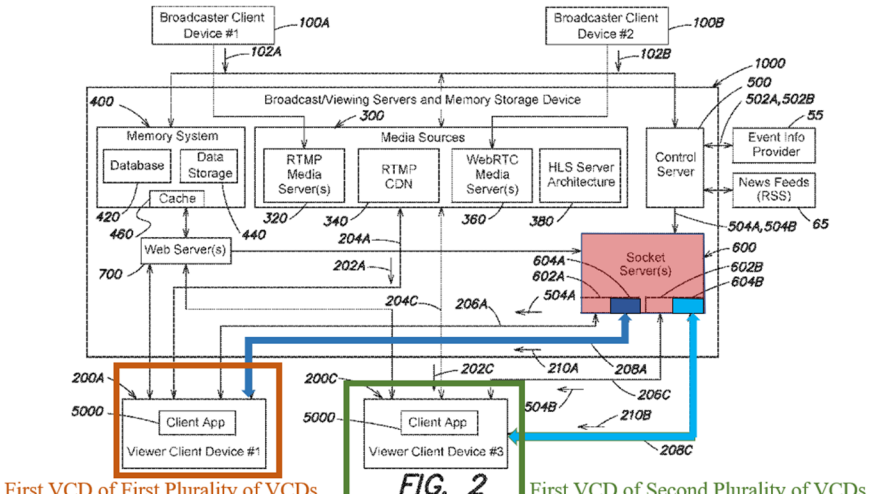
b. Dependent Claim 24

i. **Elements 24.0-24.1: "The method of claim 23, further comprising: D) transmitting and receiving first chat information regarding the first live sporting event via at least one first**

chat/system event Internet communication channel between at least one first chat/system event socket of the at least one socket server and the first viewer client device, wherein the at least one first chat/system event socket corresponds to the first chat information."

612. In my opinion, Ellis in view of Spivey discloses these elements. Ellis in view of Spivey discloses transmitting and receiving chat information regarding the first live sporting event via individual communication channels between a first socket and a first VCD. §§X.B.4.c, *supra*.

613. The '687 explains that a chat/system event communication channel may be any generic internet communication channel. Ex.1003, 5:38-45. These elements only further require that the system "transmits and receives" first "chat information regarding" the first "live sporting event " over another separate, individual connection between a first "chat/system event socket" and a first VCD:



Ex.1003, 22:17-48, Claim 24, Fig. 2. In sum, it is my opinion that a POSA would have understood that these elements require that chat information is transmitted and received over a communication channel from another socket that is dedicated for chat/system information.

614. Ellis discloses sending and receiving chat information because users "may also leave feedback regarding the commentary that other users in the media system may view when they access the same commentary." Ex.1016 ¶132. Such commentary includes "text." *Id*., ¶132. Thus, it is my opinion that a POSA would have understood that Ellis discloses a system that transmits and receives first chat information regarding the first live sporting event. *See also* Ex.1016 ¶126 ("To create and send a message (e.g., instant message or email message), the user may select send message option.").

615. Ellis also discloses providing this information on a separate channel to users. 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. *Id*. Such commentary includes "text." *Id*., ¶132. Thus, it is my opinion that a POSA would have understood that Ellis discloses transmitting via at least one first chat/system event Internet communication channel

between at least one first chat/system event socket of the at least one socket server and the first VCD.

616. Ellis in view of Spivey discloses a socket specifically designated for chat/system information. Spivey utilizes topic queues to organize "data payloads" that "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 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. 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." In sum, the statistics would have been separate from the chat information. Thus, a POSA would have understood Spivey to teach a first chat/system event socket (*e.g.*, "gamecenter.nfl.ravenssteelers.chat") corresponding to the first chat information regarding the first live sporting event.

7. '687 Ground 7: Claim 26 is Obvious Over Ellis in View of Spivey, Kellicker, and Phillips

617. For the reasons stated in §IX.C, above, it is my opinion that a POSA would have been motivated to combine the teachings of Ellis, Spivey, Kellicker and Phillips to arrive at the claimed invention with a reasonable expectation of success.

a. Dependent Claim 26

i. **Elements 26.0-26.2: "The method of claim 24, wherein: the at least one media source comprises: at least one of: at least one real-time messaging protocol (RTMP) media server to receive the first live stream of digital content; and at least one web real time communication (WebRTC) media server to receive the first live stream of digital content; and"**

618. In my opinion, Ellis in view of Spivey and Kellicker teaches these elements. The above elements are essentially the same as claim elements 10.0-10.2. Therefore, for at least the reasons explained above at §X.B.2.a.i. above, these elements would have been obvious to a POSA.

619. As explained in §IX.B above, Kellicker discloses a server (120/1250) that receives incoming live media streams (e.g., live incoming media stream (1220)), processes the incoming media streams and serves such media streams to playback devices in real-time. Ex.1049, ¶¶100-101, 106. Kellicker discloses that this server can be implemented through a plurality of such 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"), ¶24. Thus, Kellicker discloses at least one media source in the form of real-time streaming media servers.

620. Kellicker also teaches that the server(s) may support various streaming protocols including, RTMP, and WebRTC. *Id*., ¶106. ("computing device 1250 may also support real time messaging protocol (***RTMP***) (and variants thereof), real-time

streaming protocol (RTSP), real-time transport protocol (RTTP), web real-time communication (***WebRTC***), and MPEG-2 transport stream (MPEG-TS)"), ¶ 24 (disclosing transcoders/transmuxers for transforming media streams). Thus, Kellicker discloses media sources in the form of a real-time RTMP media server and/or a real-time WebRTC server.

621. Kellicker teaches providing its server(s) with the capability to "enable the computing device 1250 to process data in accordance with multiple coding technologies and protocols." *Id*., ¶103-104. In my opinion, it would have been obvious to a POSA to have implemented the content source(s) 130 of Ellis in the form of a real-time RTMP media server and/or a real-time WebRTC server. In my opinion, 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.

622. To the extent required by the claims, in my opinion it would have also been obvious to a POSA to have modified Ellis to implement the content source(s) 130 in the form of multiple servers, including a RTMP server ***and*** a WebRTC server. A POSA would have recognized that the use of multiple servers, as taught by Kellicker, would have provided the system with even more flexibility, capacity and adaptability to handle multiple different types of incoming media and comply with multiple types of protocols. *Id*., ¶98.

ii. **Element 26.3: "an HTTP live streaming (HLS) caching server architecture communicatively coupled to at least one of the at least one RTMP media server and the at least one WebRTC media server,"**

623. In my opinion, Ellis in view of Spivey and Kellicker teaches this element. The above element is essentially the same as claim element 10.3. Therefore, for at least the reasons explained above at §X.B.2.a.ii. above, this element would have been obvious to a POSA.

624. To reiterate, Kellicker teaches that the at least one RTMP media server and/or the at least one WebRTC media server are communicatively coupled to one or more other servers 1280. Ex.1049, ¶102. For example, 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., an edge server), etc. *Id*.; FIG. 12. Kellicker also explains that the data transmitted to the servers 1280 includes "media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof)." *Id*. Thus, Kellicker teaches that the RTMP media server and/or the WebRTC media server are communicatively coupled to the one or more other servers 1280.

625. Kellicker further discloses that the media streams received by the one or more other servers 1280 "may be associated with the same encoding format and

transmission protocol or may be associated with different encoding formats and transmission protocols." *Id*. and ¶24 (disclosing transcoders/transmuxers for transforming media streams). Kellicker also specifically teaches providing the real-time streaming server(s) with the HLS streaming protocol ability. *Id*. at ¶105-106. In my opinion, it would have been obvious to a POSA to implement such HLS conversion capability in the form of the one or more other servers 1280 for Ellis' content source(s). A POSA would have been motivated to provide HTTP live streaming (HLS) media streams because, as was well-known to a POSA, HLS "provides a reliable, cost-effective means of delivering continuous and long-form video over the Internet." Ex.1052, 2 (§1).

626. 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." *Id*. at ¶102. In my opinion, a POSA would have understood a CDN to be a caching architecture. Ex.1052, 2 (§1) ("[HLS] offers compatibility with large-scale HTTP caching infrastructure to support delivery to large audiences"); Ex.1055 Lepeska, 7:19-23 ("[a] content delivery network or content distribution network (CDN) is traditionally known as a system of computer containing copies of data placed at various nodes of a network, such that access to the cached data is improved by increasing bandwidth and reducing access latency"). To a POSA, the fundamental purpose of a CDN is to provide caching capability in order to improve

responsiveness to client device requests for served data (in this case streamed media. *See also,* Ex.1050.

iii. **Element 26.4-26.5: "wherein the HLS caching server architecture comprises: an HLS mother server communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server to receive, from the at least one of the at least one RTMP media server and the at least one WebRTC media server, a first HLS file suite representing the first live stream of digital content; and"**

iv. **"a plurality of HLS child servers communicatively coupled to the HLS mother server and an HLS content delivery network (HLS CDN); and"**

627. In my opinion, Ellis in view of Spivey, Kellicker, and Phillips teaches this element. The above elements are essentially the same as claim elements 11.1-11.2. Therefore, for at least the reasons explained above at §X.B.3.a.i. above, these elements would have been obvious to a POSA.

628. To reiterate, as explained above at §X.B.2., Kellicker teaches an HTTP live streaming (HLS) caching server architecture including a CDN communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server. It would have been obvious to a POSA to have provided this HLS caching server CDN 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." Ex.1050, ¶31. Consistent with the teachings of Phillips, a POSA would have been motivated to provide this server architecture in order to provide "for high-performance streaming . . . to subscribers using one or more Internet-based infrastructures." *Id*., ¶30. For example, Phillips teaches that 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 implement Phillips' 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 . . .").

629. Kellicker as modified by Phillips establishes a communication path from a first 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, coupled to an HLS CDN architecture (1282), and from the CDN to a client/user device (1270):

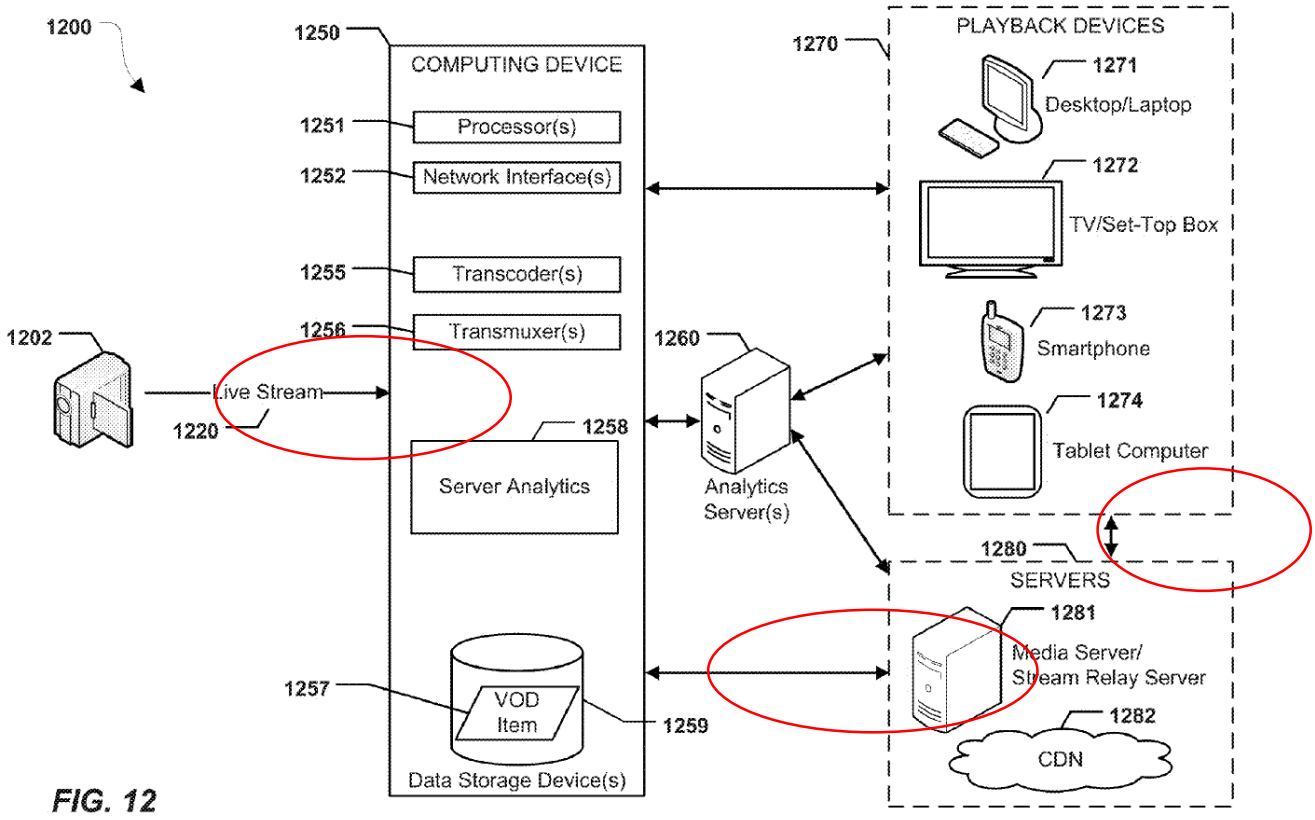


FIG. 12

Ex.1049, Fig.12 (annotated).

630. In particular, Kellicker discloses that the media server(s) 1250 is configured to transmit data to "a server of a content distribution network (CDN) 1282." Id., ¶102. Further, Kellicker discloses that, "[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.

631. In similar fashion, Phillips teaches that content that is 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 that 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 contents and/or nature of the files being handled by the system could be 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.1050, §§8.4-8.6. As can be seen, Phillips’ 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.

632. With respect to the "HLS file suite" this represents nothing more than conventional HLS file transfer protocol, as acknowledged in the ‘687. 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.1050, §§8.1-8.7. By delivering content from Kellicker’s server 1250 to a Phillips-type CDN, the HLS caching server architecture

comprises: an HLS mother server (e.g., national server, regional servers) which are communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server.

633. Hence, 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 live stream of digital content." A POSA would have understood that 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. As explained in the '687, an "HLS file suite" represents nothing more than the contents of a conventional HLS file transfer protocol. Ex.1003, 38:1–9. In my opinion, 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.1052, 27-29 (§ 8.1-8.7).

634. Alternatively, and to the extent that the nature or content of the data could be considered as limiting the claimed system, 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.1050, §§8.1-8.7. For example, as explained in Kellicker, this HLS caching server architecture receives a portion of the first HLS file suite (video stream) representing the first live stream of digital content. Ex.1049, ¶102 ("data

transmitted to the playback devices 1270 and to the servers 1280 includes media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof"). Further, Kellicker discloses that "first HTTP port(s) may be used to communicate a master rendition list and media chunks . . .," each of which is a portion of the HLS file suite. Ex.1049, ¶110; *see also* Ex.1003, 38:1–9.

635. Similarly, Phillips teaches that the first HLS file suite is received (media chunks and manifest/playlist). For example, Phillips teaches that content that is requested for delivery by the CDN may include a received "manifest" that is made available to a "content delivery node" (e.g., "mother" or "child" server) of the CDN server architecture. Ex.1050, 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"). And, to the extent that 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 (e.g., a "child" server). Ex.1050, Abstract, ¶31, Fig. 1. As another example, Phillips explains that the media portions of the HLS file suite are received by the CDN architecture. Ex.1050, FIG. 12 ("Pull Missing Segments for ABR Assets", which are media chunks of the HLS file suite), ¶9.

v. **Element 26.6: "A) comprises providing the first copy to the first viewer client device via the first video communication channel between the HLS CDN and the first viewer client device, wherein the first copy is based on the first HLS file suite received by the HLS mother server."**

636. This element is essentially the same as elements 11.3-11.4. Thus, these elements would have been obvious to a POSA for lease the same reasons set forth at §X.B.3.a.ii.-iii. Above. As explained above in §X.B.6.a.i.-ii, Ellis as modified by Spivey establishes the first video communication channel to provide the first copy of digital content to the first viewer client device. As further explained at §§X.B.2.a.iii-iv, X.B.3.a.ii-iii, Ellis as modified by Spivey, Kellicker and Phillips establishes that the first video Internet communication channels are provided between the first viewer client device and the HLS CDN. *See also* Ex.1050, ¶33 ("Each local edge server is operable to serve a plurality of subscribers in a serving area for delivering content to associated end stations or client devices, as illustrated by an example client device or user equipment (UE) device 105 that may for consume or access streaming content delivered via CDN 100 in any type or number of access technologies including broadband access via wired and/or wireless (radio) communications."), ¶28 ("client devices may also access or consume content/services provided over… packet-switched wide area public network such as the Internet via suitable service provider access networks.").

637. As explained at §X.B.3.a.i., a POSA would have understood that Ellis as modified by Spivey, Kellicker, and Phillips establishes that the first copy of the first live stream of digital content is based on the (first) HLS file suite as required by the HLS specification and the CDN distribution process.

8. '687 Ground 8: Claims 23-24 Are Obvious Over Herzog in View of Spivey and Ellis

638. For the reasons stated in §IX.F, above, it is my opinion that a POSA would have been motivated to combine the teachings of Herzog, Ellis and Spivey, to arrive at the claimed invention with a reasonable expectation of success.

a. Independent Claim 23

i. Element 23.0: "A method for providing, to a first viewer client device, a first copy of a first live stream of digital content relating to a first live sporting event, the method comprising:"

639. To the extent the preamble is limiting, it is my opinion that Herzog in view of Ellis discloses this limitation.

640. Herzog provides "live video and audio data" to user devices over a "communication network." *See* Ex.1015, Abstract. Herzog improves on traditional "sports application systems" where the meta-data is included "as part of the video broadcast." Ex.1015 ¶83, FIG. 4.

641. Herzog discloses a content provider (42) that provides a live stream of content, including video, to a solutions integrator: (44), who then provides the content to an end user device (45):

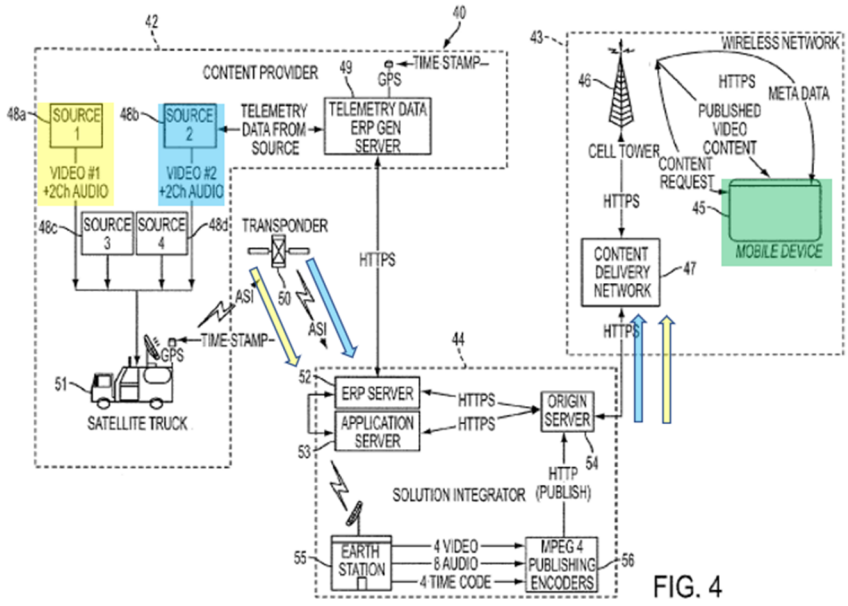


FIG. 4

642. Herzog discloses providing "multiple video/audio streams of race cars and drivers, as they are imaged at different locations on the race track:" Ex.1015 ¶45, Fig 4. Thus, Herzog discloses providing a first copy of a live stream of digital content relating to a first live sporting event to a first client device.

ii. **Element 23.1: "A) providing the first copy to the first viewer client device via a first video communication channel between at least one media source and the first viewer client device;"**

643. In my opinion Herzog discloses this limitation.

644. Herzog discloses MPEG 4 Publishing Encoders ("Encoders") as at least one media source. Ex.1015 ¶48. In my opinion—just like in the '687—these encoders receive all of the video and audio data from the live event, such as the Indy Car Race. *Id.* ¶¶46 (noting that any form of communication technologies may be used to send the video/audio data to MPEG Publishing Encoders), ¶49. The

Encoders can "encode" the video and audio data and then the "encoded video is then published to the content delivery network 47 (for example) via IP utilizing [] HLS." *Id.* ¶¶49, 55. For example, Herzog provides the selected video stream from the Encoders to each user individually via "an HTTPS Live Streaming (HLS)-based media streaming communications protocol." Ex.1015 ¶51, Figs. 1, 4.

iii. **Element 23.2: "B) retrieving, via the Internet, first event information germane to the first live sporting event; and"**

645. Herzog's ERP Server connects to the ERP Gen Server over the internet:

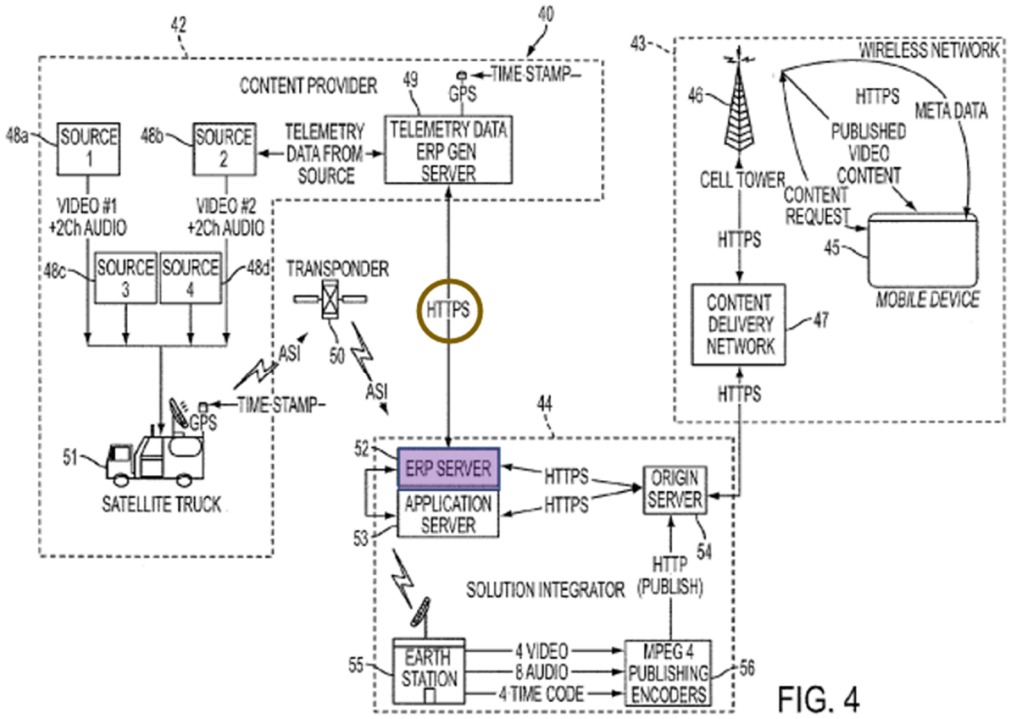


FIG. 4

Ex.1015 ¶¶45, 47, 50, Fig. 4 (annotated). The ERP Server is a generic server that retrieves meta-data from the ERP Gen Server. *Id.* ¶¶16, 47, 50. In my opinion, it contains the same components as the control server described in the '687: a

processor, memory device, and communication interfaces. *Id.* ¶13, 85-86, Fig. 8; Ex.1003, 55:15-25. The ERP Server "acts as an intermediary between the ERP generator server and the origin server." Ex.1015 ¶50.

646. The meta-data retrieved includes event information germane to the sporting event such as "the status of each car and driver in the race" and may include "lap number, car speed and acceleration" but there may be as many as "50 different parameters that are available." *Id.* ¶45. The ERP Server retrieves this first event information via the internet and from an event information provider because it retrieves the information, *e.g.*, status of each car and driver in the race, in HTTPS format, using an internet communication protocol from an ERP Server. *Id.* ¶¶17, 50.

647. Ellis's SCS also 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 . . ."). Additionally, Ellis's 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." Ex.1016 ¶¶41-42, 142. The SCS periodically retrieves event information germane to all live sporting events, *e.g.*, score information because the SCS "provide[s] real-time sports scores and game summary information, … spread and wagering data, and any other sports-related

information to user equipment 102." *Id.* ¶41; *see id.* ¶¶72, 117. In my opinion, a POSA would have understood that this information is retrieved via the internet. *Id.* ¶¶37-38, 40, 142. Furthermore, a POSA would have understood that this was how real-time data was routinely transferred to users at the time of Herzog, Ellis and Spivey—including score updates and wagering information. Ex.1016 ¶¶42, 142. Additionally, in my opinion 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.

iv. **Element 23.3: "C) transmitting the first event information to at least the first viewer client device via a first event information communication channel between at least one first event socket of at least one socket server and the first viewer client device, wherein the at least one first event socket corresponds to the first event information germane to the first live sporting event."**

648. In my opinion, Herzog in view of Ellis and Spivey discloses a socket server communicatively coupled to the control server (which, as discussed above, is disclosed by Herzog's ERP Server). Herzog discloses providing event information over a separate connection to user equipment. Ex.1015 ¶¶48-52, 67, 70-77, Figs. 4, 6. That process includes the ERP Server transmitting data to the Origin Server:

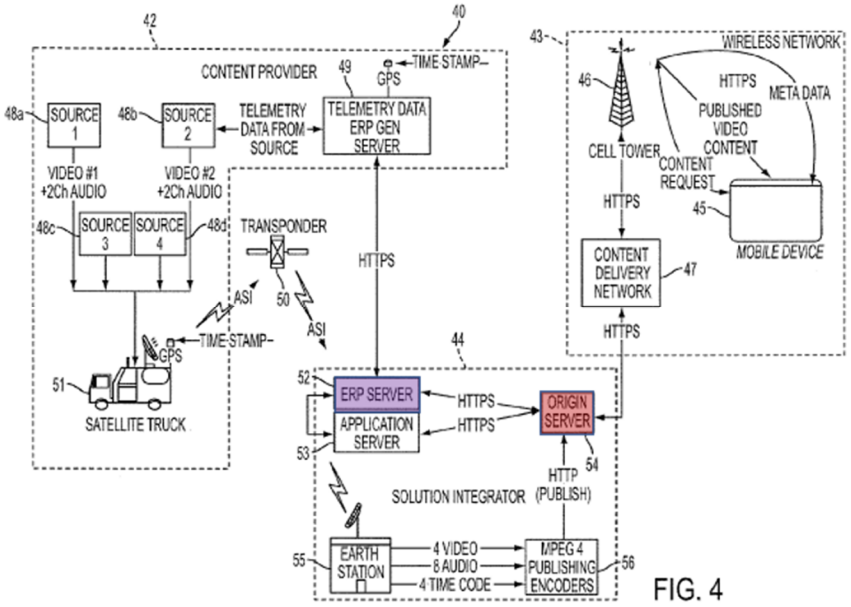


FIG. 4

Ex.1015 ¶¶50-51, 83, Fig. 4. But in a system of Spivey combined with Herzog and Ellis, it is my opinion that a POSA would have looked to Spivey to add additional backend architecture to organize the meta-data from multiple events and reduce latency in the real-time data delivery system for at least the same reasons set forth in §X.6.a.iv. I believe a POSA would have done this by incorporating Spivey's MQSD into at least Herzog's Origin Server.

649. Like the '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.

650. Moreover, in my opinion a POSA would have understood that Spivey's MQSD carries out the same purpose as the '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." *Id.* ¶36; *see id.* ¶¶45 ("[MQS] device 310 may be a representative example of message queue server 120"), ¶54 (describing that 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:"

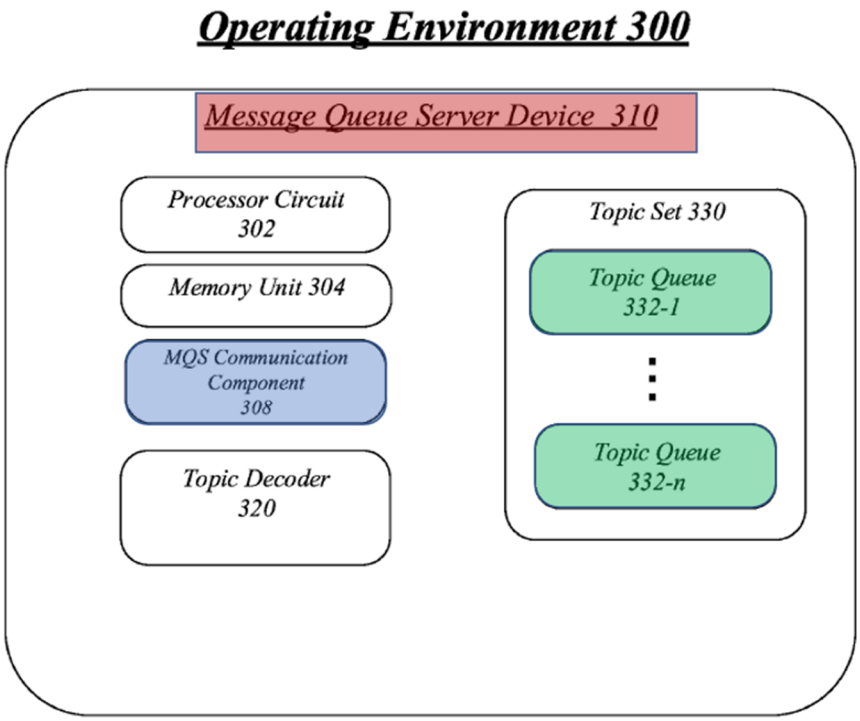


FIG. 3

Id. ¶55, Fig. 3 (annotated). MQSD contains the MQS communication component which would allow the MQSD to communicate with the ERP Server, as contemplated in Spivey. *See id.* ¶¶46 (disclosing that communication component 308 transmits and receives data), ¶54 (disclosing that communication component

408 is analogous to 308), ¶55 ("LDS communication component 418 … allows [LDSD] to … transmit the data payloads to [MQSD]."); Ex.1015, Fig. 4.

651. After receiving the data, 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. After sorting the data into the proper topic queues, the MQSD shares the information "substantially simultaneously" to "the subscribed connected client devices." *See id.* ¶96.

652. Spivey teaches that latency associated with the broadcast of event data, such as sports scores, to clients can be reduced by "using topic cues and open duplex communication connections." *Id.* ¶¶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."). As an example of such "open duplex communication connections" Spivey discloses that communications between a server and a client 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"); *see also*, *Id.* ¶¶ 43, 61, 74, and 92; claim 8. Thus, it would also have been obvious to a POSA in view of these teachings of Spivey 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. As explained above, a POSA would have recognized that doing so would further the express goals of Spivey with respect to reduction of latency in such transmissions. In my opinion, a POSA would have had a reasonable expectation of success implementing this web socket functionality in the MQSD because this would have involved nothing more than a combination of well-known, compatible elements to achieve a predictable result.

(A) First event socket corresponds to the first event information germane to the first live sporting event

653. For at least the same reasons set forth in §X.6.a.iv., Herzog as modified by Spivey in view of Ellis satisfies this aspect of the claimed invention. Spivey often discusses a connection between the client device and the system using a live data access server ("LDAS"), however, Spivey also discloses and teaches that the message queue server device may be "accessible to clients over the network directly." Ex.1017 ¶¶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 disclose transmitting first event information on"), 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. Also, the system does not require an LDAS, only that it "may include a[n] [LDAS]." *Id.* ¶38. Thus, it is my opinion that a POSA would have

understood Spivey to disclose and teach a direct connection between the client devices and MQSD. In my opinion, a POSA would have used this implementation because reducing the amount of server transfers would reduce latency, which is Spivey's main purpose. Furthermore, it was a well-known alternative to a POSA to implement the functionality of multiple servers as a single server. Ex.1016, ¶¶4-7.

654. In my opinion, a POSA would have understood that 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. *Id.* It is my opinion that 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. *See* Ex.1003 at 20:55-21:4, 21:63-22:16; Ex.1029, pp. 14-20; Ex1030; §IV.C.3., *supra*.

655. In the same way, 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 then 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.

656. 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. *See 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:

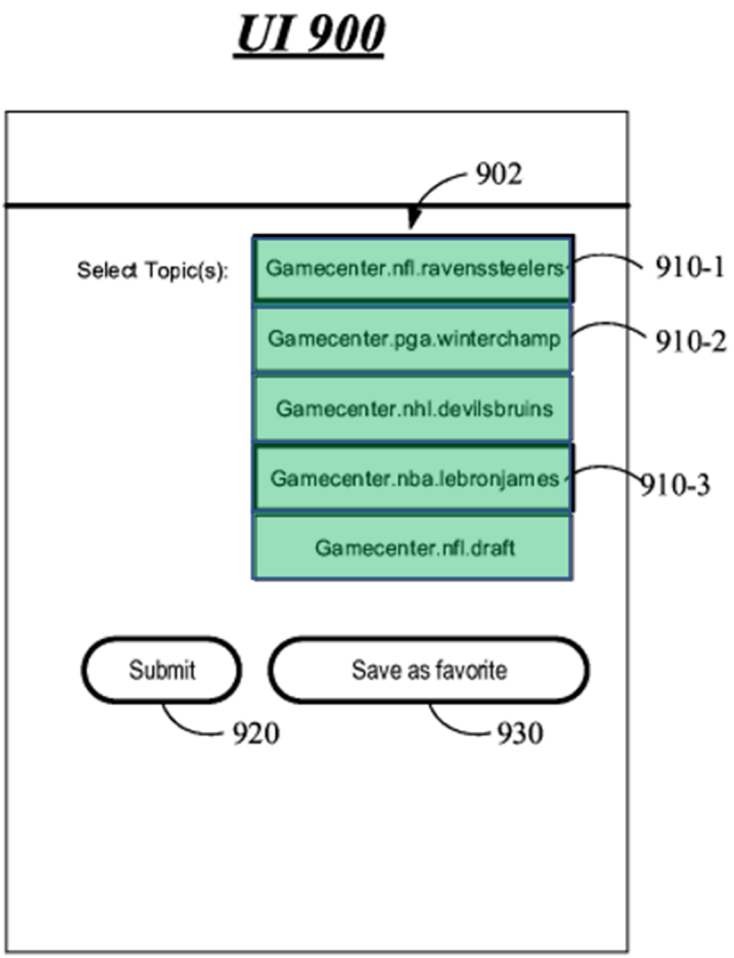


FIG. 9

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, it is my opinion that a POSA would have understood that Spivey's topic queues disclose the '687's first event sockets corresponding to the first event information germane to the first live sporting event.

657. Alternatively, it would also have been obvious to a POSA in view of these teachings of Spivey 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 in §IX.A.1, IX.F and §X.B.1.a.vi.

(B) Transmitting the first event information to at least the first VCD

658. For at least the same reasons set forth in §X.6.a.iv., Herzog as modified by Spivey in view of Ellis satisfies this aspect of the claimed invention. In my opinion, Herzog discloses a first VCD. *See* §X.B.8.a.i-iii., *supra.* In my opinion, a POSA would have understood that both Herzog and Spivey further disclose transmitting the first event information to the first VCD. 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, 48 ("data payload may include, for example, a score, a play, an error, an injury, a player at-bat, and so forth"). In Herzog, the Origin Server may be configured to transmit "the meta-data to the

content delivery network 47," which includes car position, and the user receives "in an HTTPS communications protocol." Ex.1015 ¶¶45, 51.

(C) Transmitting via a first event information communication channel between at least one first event socket of the at least one socket server and the first VCD

659. For at least the same reasons set forth in §X.6.a.iv., Herzog as modified by Spivey in view of Ellis satisfies this aspect of the claimed invention. As previously noted, Herzog discloses a first VCD. *See* §X.B.8.a.i-ii.

660. In my opinion, Herzog in view of Spivey and Ellis 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 Internet." *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, Herzog discloses providing event information over a separate communication path than the video signal where there is "a unique video and meta-data source that do not follow the same path of transmission:"

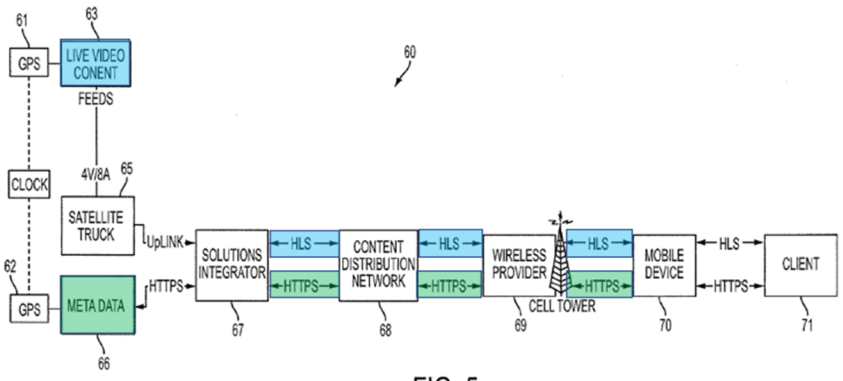


FIG. 5

Ex.1015 ¶47-51, 83, Figs. 1, 4, 5 (annotated) 6. Thus, it is my opinion that a POSA would have understood Spivey to disclose transmitting first event information via a separate internet connection. In my opinion, a POSA would have known, and considered obvious, that the websocket client-server communication protocol employed by Spivey was an alternative to Herzog's HTTP polling. §IV.C

661. Further, in my opinion this information is provided via a connection between a topic queue and the first user equipment device. Users can access the MQSD using a URL. *See* Ex.1017 ¶¶81-82. Then, the user can access the specific topic queue she wishes to receive information from by using a text string, *e.g.*, a topic label or Event ID, associated with a topic queue. *Id.* ¶179, Fig. 9 ("Gamecenter.nfl.ravenssteelers."). Thus, it is my opinion that a POSA would have understood that Herzog in view of Ellis and Spivey discloses distributing event information from a socket to a user device over an internet connection. Further, I believe that 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.

b. Dependent Claim 24

i. **Elements 24.0-24.1: "The method of claim 23, further comprising: D) transmitting and receiving first chat information regarding the first live sporting event via at least one first chat/system event Internet communication channel between at least one first chat/system event socket of the at least one socket server and the first viewer client device, wherein the at least one first chat/system event socket corresponds to the first chat information."**

662. In my opinion, Herzog in view of Ellis and Spivey discloses these elements. Herzog in view of Ellis and Spivey discloses transmitting information via individual communication channels between a socket and a first VCD. §X.B.8.a *supra*.

663. The '687 explains that a chat/system event communication channel may be any generic internet communication channel. Ex.1003, 5:38-45. These elements only further require that the system "transmits and receives" first "chat information regarding" the first "live sporting event" over another separate, individual connection between a first "chat/system event socket" and a first VCD:

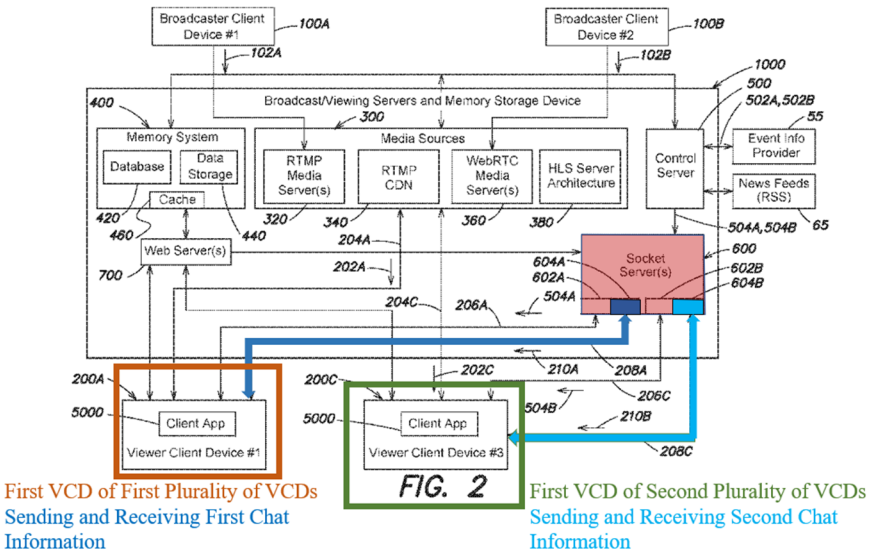


FIG. 2

Ex.1003, 22:17-48, Claim 24, Fig. 2. In sum, it is my opinion that a POSA would have understood that these elements require that chat information is transmitted and received over a communication channel from another socket that is dedicated for chat/system information.

664. The '687 describes chat information as communications "between a given broadcaster … and corresponding viewers … of chat messages." Ex.1003, 20:22-27. Further, "viewers of the same live event may" view different video feeds of the same live event and "participate in different chat information exchanges … and all share the same event information in a synchronized manner." *Id.,* 23:17-28. Additionally, the '687's explanation of chat/system event sockets indicate that they are the same as the other sockets, *i.e.*, a communication endpoint on a server that is uniquely addressable, *e.g.*, by using an identifier (StreamID) that may be contained,

e.g., in a URL. *Id.*, 20:55-21:4, 21:63-22:16. The only difference is that these sockets store and send chat messages, not "event data." *See id.,* 23:55-24:15.

665. Herzog as modified by Ellis and Spivey render these elements obvious for lease the same reasons set forth above at §X.6.b.i. Ellis discloses chat information because users "may also leave feedback regarding the commentary that other users in the media system may view when they access the same commentary." Ex.1016 ¶132. Thus, a POSA would have understood that Herzog in view of Ellis and Spivey discloses a system that transmits and receives chat information regarding the live stream of digital content. *Id.* ¶126 ("To create and send a message (e.g., instant message or email message), the user may select send message option.").

666. Ellis, consistent with Herzog's separate meta-data stream, also discloses providing this event data and chat information on a separate channel to users. Ex.1015 ¶¶49-53, 59-61, Fig. 5; Ex.1016 ¶¶39-41, Fig. 1. 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. Consistent with Herzog, Ellis provides that in "some embodiments, this sports-related information is provided to user equipment 102 as a plurality of real-time data feeds." Ex.1016 ¶41. Thus, A POSA would have understood that Herzog in view Ellis of Spivey discloses transmitting via at least one

first chat/system event Internet communication channel between at least one first chat/system event socket of the at least one socket server and the first VCD.

667. Herzog in view of Ellis and Spivey discloses a socket specifically designated for chat information. Spivey utilizes topic queues to organize "data payloads" that "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." 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. For example, if there were two topics "at the same hierarchical level" the data payloads placed in "'gamecenter.nfl.ravenssteelers.statistics' would not be placed in" another topic queue "gamecenter.nfl.ravenssteelers.plays." *Id*. In sum, the statistics would be provided separately from the other information, e.g., chat information. Thus, a POSA would have understood Spivey to teach a first chat/system event socket (*e.g.*, "gamecenter.nfl.ravenssteelers.chat") corresponding to the first chat information.

9. '687 Ground 9: Claim 26 is Obvious Over Herzog in View of Spivey, Ellis, Kellicker, and Phillips

668. For the reasons stated in Section IX.G, above, it is my opinion that a POSA would have been motivated to combine the teachings of Herzog, Spivey,

Ellis, Kellicker, and Phillips to arrive at the claimed invention with a reasonable expectation of success.

i. **Elements 26.0-26.2: "The method of claim 24, wherein: the at least one media source comprises: at least one of: at least one real-time messaging protocol (RTMP) media server to receive the first live stream of digital content; and at least one web real time communication (WebRTC) media server to receive the first live stream of digital content; and"**

669. In my opinion, Herzog in view of Ellis, Spivey and Kellicker teaches this element, for at least the same reasons set forth in §X.B.7.a.i..

670. As explained in §IX.B above, Kellicker discloses a server (120/1250) that receives incoming live media streams (e.g., live incoming media stream (1220)), processes the incoming media streams and serves such media streams to playback devices in real-time. Ex.1049, ¶¶100-101, 106. Kellicker discloses that this server can be implemented through a plurality of such 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"), ¶24. Thus, Kellicker discloses at least one media source in the form of real-time streaming media servers.

671. Kellicker also teaches that the server(s) may support various streaming protocols including, RTMP, and WebRTC. *Id*., ¶106. ("computing device 1250 may also support real time messaging protocol (***RTMP***) (and variants thereof), real-time streaming protocol (RTSP), real-time transport protocol (RTTP), web real-time

communication (***WebRTC***), and MPEG-2 transport stream (MPEG-TS)"), ¶ 24 (disclosing transcoders/transmuxers for transforming media streams). Thus, Kellicker discloses at least one media source in the form of a real-time RTMP media server and/or a real-time WebRTC server.

672. Kellicker teaches providing its server(s) with the capability to "enable the computing device 1250 to process data in accordance with multiple coding technologies and protocols." *Id*., ¶103-104. In my opinion, it would have been obvious to a POSA to have implemented the media sources (i.e., "encoders" 56) of Herzog, at least in part, in the form of a real-time RTMP media server and/or a real-time WebRTC server. In my opinion, a POSA would have been motivated to do so in order to provide the content source of Herzog with the processing and transmission ability to conform with these well-known and widely utilized protocols. *Id*., ¶¶103-106.

673. To the extent required by the claims, in my opinion it would have also been obvious to a POSA to have modified Herzog to implement the media sources 56 in the form of multiple servers, including a RTMP server ***and*** a WebRTC server. A POSA would have recognized that the use of multiple servers, as taught by Kellicker, would have provided the system with even more flexibility, capacity and adaptability to handle multiple different types of incoming media and comply with multiple types of protocols. *Id*., ¶98.

ii. **Element 26.3: "an HTTP live streaming (HLS) caching server architecture communicatively coupled to at least one of the at least one RTMP media server and the at least one WebRTC media server,"**

674. In my opinion, Herzog in view of Ellis, Spivey and Kellicker teaches this element, for at least the same reasons set forth in §X.B.7.a.ii.

675. Kellicker teaches that the at least one RTMP media server and/or the at least one WebRTC media server are communicatively coupled to one or more other servers 1280. Ex.1049, ¶102. For example, 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., an edge server), etc. *Id*.; FIG. 12. Kellicker also explains that the data transmitted to the servers 1280 includes "media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof)." *Id*.

676. Kellicker further discloses that the media streams received by the one or more other servers 1280 "may be associated with the same encoding format and transmission protocol or may be associated with different encoding formats and transmission protocols." *Id*. and ¶24 (disclosing transcoders/transmuxers for transforming media streams). Kellicker also specifically teaches providing the real-time streaming server(s) with the HLS streaming protocol ability. *Id*. at ¶105-106.

In my opinion, it would have been obvious to a POSA to implement such HLS conversion capability in the form of the one or more other servers 1280, such as the media/relay server 1281, for the Herzog's encoders 56 (media sources). A POSA would have been motivated to provide HTTP live streaming (HLS) media streams because, as was well-known to a POSA, HLS "provides a reliable, cost-effective means of delivering continuous and long-form video over the Internet." Ex.1052, 2 (§1).

677. 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." *Id*. at ¶102. In my opinion, a POSA would have understood a CDN to be a caching architecture. Ex.1052, 2 (§1) ("[HLS] offers compatibility with large-scale HTTP caching infrastructure to support delivery to large audiences"); Ex.1055 Lepeska, 7:19-23 ("[a] content delivery network or content distribution network (CDN) is traditionally known as a system of computer containing copies of data placed at various nodes of a network, such that access to the cached data is improved by increasing bandwidth and reducing access latency"). To a POSA, the fundamental purpose of a CDN is to provide caching capability in order to improve responsiveness to client device requests for served data (in this case streamed media. *See also,* Phillips (Ex.1050). Thus, in my opinion, a POSA would have understood

that Kellicker teaches an HLS caching server architecture communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server.

iii. **Element 26.4-26.5: "wherein the HLS caching server architecture comprises: an HLS mother server communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server to receive, from the at least one of the at least one RTMP media server and the at least one WebRTC media server, a first HLS file suite representing the first live stream of digital content; and"**

iv. **"a plurality of HLS child servers communicatively coupled to the HLS mother server and an HLS content delivery network (HLS CDN); and"**

678. In my opinion, Herzog in view of Ellis, Spivey, Kellicker, and Phillips teaches this element for at least the same reasons explained at X.B.7.a.iii.-iv.

679. As explained above at §X.B.2., Kellicker teaches an HTTP live streaming (HLS) caching server CDN architecture including a CDN communicatively coupled to at least one RTMP media server and/or at least one WebRTC media server. It would have been obvious to a POSA to have provided this HLS caching server CDN 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." Ex.1050, ¶31. Consistent with the teachings of Phillips, a POSA would have been motivated to provide this server architecture

in order to provide "for high-performance streaming . . . to subscribers using one or more Internet-based infrastructures." Id., ¶30. For example, Phillips teaches that 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 implement Phillips' 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 . . .").

680. Kellicker as modified by Phillips establishes a communication path from a first 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, coupled to an HLS CDN architecture (1282), and from the CDN to a client/user device (1270):

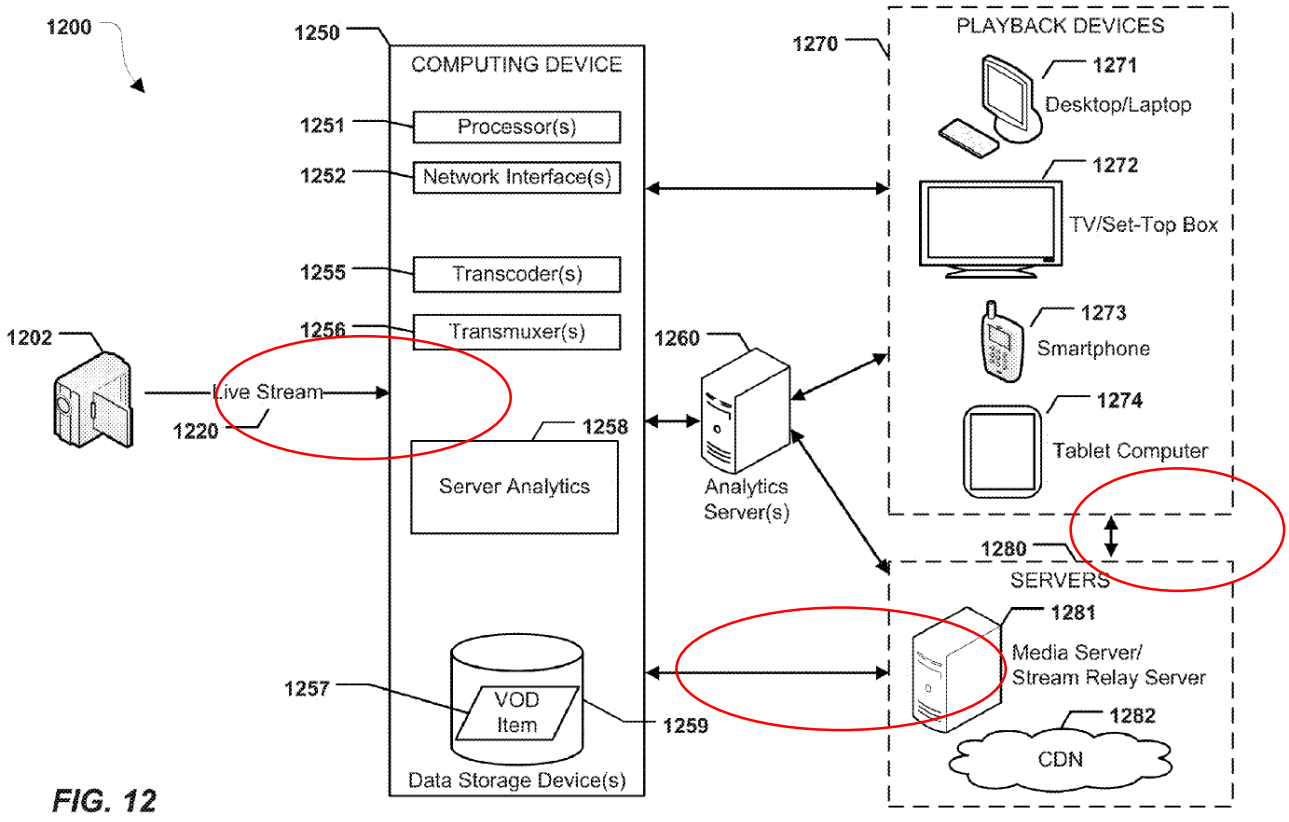


FIG. 12

Ex.1049, Fig.12 (annotated).

681. In particular, Kellicker discloses that the media server(s) 1250 is configured to transmit data to "a server of a content distribution network (CDN) 1282." Id., ¶102. Further, Kellicker discloses that, "[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.

682. In similar fashion, Phillips teaches that content that is 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 that 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 contents and/or nature of the files being handled by the system could be 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.1050, §§8.4-8.6. As can be seen, Phillips’ 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.

683. With respect to the "HLS file suite" this represents nothing more than conventional HLS file transfer protocol, as acknowledged in the ‘687. 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.1050, §§8.1-8.7. By delivering content from Kellicker’s server 1250 to a Phillips-type CDN, the HLS caching server architecture

comprises: an HLS mother server (e.g., national server, regional servers) which are communicatively coupled to the at least one of the at least one RTMP media server and the at least one WebRTC media server.

684. Hence, 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 live stream of digital content." A POSA would have understood that 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. As explained in the '687, an "HLS file suite" represents nothing more than the contents of a conventional HLS file transfer protocol. Ex.1003, 38:1–9. In my opinion, 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.1052, 27-29 (§ 8.1-8.7).

685. Alternatively, and to the extent that the nature or content of the data could be considered as limiting the claimed system, 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.1050, §§8.1-8.7. For example, as explained in Kellicker, this HLS caching server architecture receives a portion of the first HLS file suite (video stream) representing the first live stream of digital content. Ex.1049, ¶102 ("data

transmitted to the playback devices 1270 and to the servers 1280 includes media streams (e.g., streams including media content, such as audio, video, text, still images, or a combination thereof"). Further, Kellicker discloses that "first HTTP port(s) may be used to communicate a master rendition list and media chunks . . .," each of which is a portion of the HLS file suite. Ex.1049, ¶110; *see also* Ex.1003, 38:1–9.

686. Similarly, Phillips teaches that the first HLS file suite is received (media chunks and manifest/playlist). For example, Phillips teaches that content that is requested for delivery by the CDN may include a received "manifest" that is made available to a "content delivery node" (e.g., "mother" or "child" server) of the CDN server architecture. Ex.1050, 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"). And, to the extent that 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 (e.g., a "child" server). Ex.1050, Abstract, ¶31, Fig. 1.

687. As another example, Phillips explains that the media portions of the HLS file suite are received by the CDN architecture. Ex.1050, FIG. 12 ("Pull

Missing Segments for ABR Assets", which are media chunks of the HLS file suite), ¶9.

v. **Element 26.6: "A) comprises providing the first copy to the first viewer client device via the first video communication channel between the HLS CDN and the first viewer client device, wherein the first copy is based on the first HLS file suite received by the HLS mother server."**

688. This element is satisfied for at least the same reasons set forth at §X.B.7.a.v. As explained above in §X.B.8.a.ii, Herzog as modified by Ellis and Spivey establishes the first video communication channel to provide the first copy of digital content to the first viewer client device. As further explained at §§X.B.2.a.iii-iv, X.B.3.a.ii-iii, and X.B.9.a.i-iv Herzog as modified by Ellis, Spivey, Kellicker and Phillips establishes that the first video Internet communication channels are provided between the first viewer client device and the HLS CDN. *See also* Ex.1050, ¶33 ("Each local edge server is operable to serve a plurality of subscribers in a serving area for delivering content to associated end stations or client devices, as illustrated by an example client device or user equipment (UE) device 105 that may for consume or access streaming content delivered via CDN 100 in any type or number of access technologies including broadband access via wired and/or wireless (radio) communications."), ¶28 ("client devices may also access or consume content/services provided over… packet-switched wide area public network such as the Internet via suitable service provider access networks.").

689. As explained at §X.B.3.a.i.-ii and X.B.9.i.-iv., Herzog as modified by Ellis, Spivey, Kellicker, and Phillips establishes that the first copy of the first live stream of digital content is based on the (first) HLS file suite as required by the HLS specification and the CDN distribution process.

XI. SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS

690. I have been advised that Patent Owner has alleged that secondary considerations demonstrate the nonobviousness of the '697 and '687 patents. However, I have also been advised that for such evidence to be given substantial effect, that Patent Owner is required to establish a nexus to the claimed inventions. I have further been advised that to date, Patent Owner has failed to do so. Thus, these assertions fail to overcome the combinations of prior art cited above in showing the obviousness of claims 1, 10-18, 23-24, and 26 of the '697 patent or claims 1, 10-18, 23-24, and 26 of the '687 patent. The sheer number of independently developed, simultaneous inventions, developed within in a relatively short time-period are persuasive indicia of obviousness. However, should Patent Owner assert any secondary considerations of non-obviousness, I reserve the right to respond.

XII. CONCLUSION

691. For all the reasons discussed above, it is my opinion that claims 1, 10-18, 23-24, and 26 of the '697 patent and claims 1, 10-18, 23-24, and 26 of the '687 patent are invalid.

692. I hereby declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct, and that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I understand that willful false statements are punishable by fine or imprisonment or both. *See* 18 U.S.C. §1001.

Date: December 13, 2024

Respectfully submitted,

Dr. Omid Kia, Ph.D.