UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Genius Sports Ltd.

Petitioner,

v.

SportsCastr Inc. (d/b/a PANDA Interactive)

Patent Owner.

Case IPR2024-01309

Patent No. 10,425,697

DECLARATION OF AVIEL RUBIN, Ph.D.
IN SUPPORT OF PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

I. QUALIFICATIONS 2

II. UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES 8

III. STATE OF THE PRIOR ART 10

A. Internet Communication Channels 10

1. *Network Layer (Layer 3): Internet Protocol Addresses* 12

2. *Transport Layer (Layer 4): Ports and protocols like TCP, UDP* 14

3. *Data transmission with IP Addresses and Ports* 16

4. *The Client-Server Model in Network Transmissions* 17

5. *Content Delivery Networks (CDNs)* 19

B. Real Time Content Sharing Protocols 20

1. *Persistent/Continuous Streaming Transmission Protocols* 21

2. *Adaptive Bitrate (ABR) Protocols* 23

C. Live Audio/Video Streaming Services in 2016 26

IV. The ’697 Patent 31

A. Summary Of The ’697 Patent 31

B. Claim Construction 42

1. “means for transmitting…” 42

C. Level Of Ordinary Skill In The Art 44

V. OVERVIEW OF PRIOR ART 46

A. Overview Of Ellis 46

B. Overview Of Spivey 60

C. Overview Of Herzog 74

VI. THE CITED PRIOR ART DOES NOT DISCLOSE OR RENDER OBVIOUS ALL LIMITATIONS OF INDEPENDENT CLAIMS 19, 23, AND 27 90

A. Ground 1 Does Not Disclose Or Render Obvious All Limitations of Claims 19, 23, and 27 90

1. A POSITA would not have been motivated to combine the teachings of Ellis and Spivey 91

2. The asserted combination does not disclose or render obvious an "*event socket*" and/or *means for transmitting the first score information therethrough* or transmitting score information "*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.*" 115

B. Ground 2 Does Not Disclose Or Render Obvious All Limitations of Claims 19, 23, and 27 131

1. A POSITA would not have been motivated to combine the teachings of Herzog and Spivey 132

2. A POSITA would not have been motivated to combine the teachings of Herzog and Ellis 154

3. The asserted combination does not disclose or render obvious an "*event socket*" and/or *means for transmitting the first score information therethrough* or transmitting score information "*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*" 156

VII. CLAIMS AND/OR LIMITATIONS NOT ADDRESSED IN THIS DECLARATION 158

VIII. SECONDARY OBJECTIVE INDICIA OF NONOBVIOUSNESS 158

IX. CONCLUSION.....162

EXHIBIT LIST

Exhibit	Description
EX-2001	Declaration of Aviel Rubin, Ph.D.
EX-2002	*Curriculum vitae* of Aviel Rubin, Ph.D.
EX-2003	Joint Claim Construction Statement, Dkt. 69
EX-2004	Declaration of Dr. Stephen Wicker, September 13, 2024
EX-2005	Exhibit Not Used
EX-2006	Yahoo Finance Tanner Simkins, April 20, 2017
EX-2007	NBA.com, Steve Ashburner April 27, 2017 Article
EX-2008	Sportstechie, Robert Roble, April 22, 2017
EX-2009	Sports Tech Tokyo Award; 2019 Yahoo! Sports Tech Awards; Sports Business Awards
EX-2010	Various Media Reports on the Patented Invention
EX-2011	CNBC, December 20, 2018
EX-2012	https://sportscastr.com/about
EX-2013	https://pandainteractive.com/patents
EX-2014	https://sportscastr.com
EX-2015	https://studiopanda.live/
EX-2016	Genius Sports 2-24-18 Email
EX-2017	Genius Sports 5-7-18 Email
EX-2018	’697 Patent Genius Sports Infringement Claim Chart
EX-2019	https://www.opensourceforu.com/2015/03/a-guide-to-using-raw-sockets
EX-2020	https://learn.microsoft.com/en-us/windows/win32/winsock/tcp-ip-raw-sockets-2
EX-2021	Oracle documentation in 2011: Anatomy of the Client/Server Model. https://docs.oracle.com/cd/E13203_01/tux-edo/tux80/atmi/intbas3.htm
EX-2022	Peng, Gang. "CDN: Content distribution network." arXiv pre-print cs/0411069 (2004) https://arxiv.org/pdf/cs/0411069
EX-2023	The History of Streaming Media. https://boyetblog.s3.amazo-naws.com/PCPlus/324.Streaming.pdf
EX-2024	RFC 2326: Real Time Streaming Protocol (RTSP) https://www.ietf.org/rfc/rfc2326.txt
EX-2025	Adobe’s Real Time Messaging Protocol https://rtmp.veris-kope.com/pdf/rtmp_specification_1.0.pdf
EX-2026	WebRTC 1.0: Real-time Communication Between Browsers

Exhibit	Description
	https://www.w3.org/TR/2011/WD-webrtc-20111027/
EX-2027	HTTP Live Streaming: draft-pantos-http-live-streaming-01 https://datatracker.ietf.org/doc/html/draft-pantos-http-live-streaming-01
EX-2028	Microsoft To Add Smooth Streaming in IIS7 https://virtualizationreview.com/articles/2008/11/04/microsoft-to-add-smooth-streaming-in-iis7.aspx
EX-2029	HTTP Dynamic Streaming. (Archived May 14, 2010) https://web.archive.org/web/20100514170442/http://www.adobe.com/products/httpdynamicstreaming/
EX-2030	MPEG ratifies its draft standard for DASH. (Archived December 14, 2011) https://web.archive.org/web/20111214104626/http://mpeg.chiariglione.org/meetings/geneva11-1/geneva_press.htm
EX-2031	Mogulus Rebrands With A Killer Domain: Livestream.com https://techcrunch.com/2009/05/18/mogulus-gets-a-killer-domain-for-rebranding-livestreamcom/
EX-2032	How to watch Obama's inauguration (and the parties) online https://www.cnet.com/tech/tech-industry/how-to-watch-obamas-inauguration-and-the-parties-online/
EX-2033	Twitch is 4th in Peak US Internet Traffic https://blog.twitch.tv/en/2014/02/05/twitch-is-4th-in-peak-us-internet-traffic-90b1295af358/
EX-2034	Twitch Engineering: An Introduction and Overview https://blog.twitch.tv/en/2015/12/18/twitch-engineering-an-introduction-and-overview-a23917b71a25/
EX-2035	*Proceedings of the 25th ACM workshop on network and operating systems support for digital audio and video*, pp. 55-60, 2015 https://arxiv.org/pdf/1502.04666
EX-2036	Twitch Engineering: An Introduction and Overview https://blog.twitch.tv/en/2015/12/18/twitch-engineering-an-introduction-and-overview-a23917b71a25/

I, Aviel Rubin, Ph.D., declare as follows:

1. I have been engaged by King & Spalding LLP on behalf of SportsCastr Inc. (d/b/a PANDA Interactive) ("Patent Owner"), to provide this Declaration concerning the technical subject matter relevant to the petition for *Inter Partes* Review 2024-01039 ("Petition") of U.S. Patent 10,425,697 ("the '697 patent").

2. I am over 18 years of age. I have personal knowledge of the facts stated in this Declaration and could testify competently to them if asked to do so.

3. I have reviewed and am familiar with the specification and the claims of the '697 patent. In general, I will cite to the specification of a United States patent using the following formats: (Exhibit No., Col:Line Number(s)) or (Exhibit No., Paragraph Number(s)). For example, the citation (EX-1001, 1:1–10) points to the '697 patent specification at column 1, lines 1–10. Also, for convenience, I use italics to denote limitations from the challenged claims.

4. All the opinions in this Declaration are based on the documents I reviewed and my knowledge, experience, training and professional judgment. In forming the opinions expressed in this Declaration, I reviewed the Petition for *Inter Partes* Review of U.S. Patent No. 10,425,697 and supporting exhibits.

5. I have been asked to provide my technical opinions regarding how a person of ordinary skill in the art would have understood the claims of the '697 patent at the time of the alleged invention. I have also been asked to provide my

technical opinions on how concepts in the ’697 patent specification relate to claim limitations of the ’697 patent. In reaching the opinions provided herein, I have considered the ’697 patent, its prosecution history, and the references cited herein and have drawn as appropriate on my own education, training, research, knowledge, and personal and professional experience.

I. QUALIFICATIONS

6. In formulating my opinions, I have relied on my knowledge, training, and experience in the relevant field, which I will summarize briefly. A more detailed summary of my background, experience, and publications is set forth in my curriculum vitae (CV), which is submitted as Exhibit 2002.

7. I have more than 30 years of experience in the fields of computer science with a focus on security and computer networking. My professional career has been dedicated to issues relating to software, networks, and computer systems more generally. I have received funding from the federal government for research related to applications such as healthcare IT, border gateway routing, Internet of Things (IoT), and electronic voting.

8. I received a Bachelor of Science in Computer Science with Honors from the University of Michigan in 1989.

9. I received a Master of Science in Computer Science and Engineering from the University of Michigan in 1991.

10. I received a Ph.D. in Computer Science and Engineering from the University of Michigan in 1994. My Ph.D. dissertation was titled "Nonmonotonic Cryptographic Protocols" and concerned authentication in long-running network operations.

11. I am a Professor Emeritus in Computer Science at Johns Hopkins University (the "University"). I joined the Computer Science faculty at the University in 2003 and was promoted to full professor with tenure in April 2004. As a member of the faculty, I perform research, teach graduate and undergraduate courses in computer science and related subjects, advise undergraduate and Masters students, and supervise the research of Ph.D. candidates and other students. I served as the chair of the curriculum committee for the Computer Science department for two years. I have taught courses including Networking, Security and Privacy in Computing, Cryptography, and Advanced Topics in Computer Security. In September 2023, I was appointed Professor Emeritus. The Computer Networks course, which I developed for the department was a relatively large undergraduate class with over 70 students that included the basic fundamentals of how networks operate, TCP/IP, routing, OSI layers, performance and constraints. The students had to solve hands-on lab assignments that I created.

12. I also served as the Technical Director of the University's Information Security Institute ("ISI"). The ISI is the University's focal point for research and

education in information security, assurance, and privacy. The University, through the ISI's leadership, has been designated as a Center of Academic Excellence in Information Assurance by the National Security Agency and leading experts in the field.

13. I also have significant industry experience. I am currently the Founder and Managing Partner of Harbor Experts, which provides consulting for high-tech litigation. Since 2011, I have served as the Founder and Chief Scientist of Harbor Labs, a software and networking consulting firm specializing in medical device security. Previously, from 2005 to 2011, I served as the founder and President of Independent Security Evaluators ("ISE"), a computer security consulting firm. In that capacity, I guided ISE through the qualification process to become an independent testing laboratory for Consumer Union, which produces the Consumer Reports magazine. ISE provided consulting services to various companies on potential security flaws and weaknesses in products and systems. These companies included Fortify Software, Verdasys, and Security First Corp. As an independent testing laboratory for Consumer Union, I managed an annual project where we tested popular anti-virus products. Our results were published in Consumer Reports each year for three consecutive years.

14. In addition, I came to the University from AT&T Labs, Secure Systems Research Department, where I spent six years focused on Internet and computer

security. Prior to AT&T Labs, I spent two years at Bellcore in its Cryptography and Network Security Research group, also focusing on Internet and computer security issues. I also interned at IBM in 1989. During my time there, I worked on the IBM System/360 family of mainframe computer systems.

15. I am the author or co-author of five books in the Computer Science field: Brave New Ballot, Random House, 2006; Firewalls and Internet Security (second edition), Addison Wesley, 2003; White-Hat Security Arsenal, Addison Wesley, 2001; Peer-to-Peer, O'Reilly, 2001; and Web Security Sourcebook, John Wiley & Sons, 1997.

16. I have received numerous awards for my work and research in the field of computer security, including:

- In 2020, I received the Distinguished Service Award from the Institute of Electrical and Electronics Engineers ("IEEE") Computer Society Technical Committee on Security and Privacy;
- In 2010, I received a Fulbright Scholarship to serve as a Visiting Professor at Tel Aviv University in Israel;
- In 2009, I received a Google Research Award for my work on securing medical records on smartphones;

- In 2007, I received the Privacy Enhancing Technologies Award for Outstanding Research for my work on a security analysis of a cryptographically enabled RFID device;
- In 2004, I was named Baltimorean of the Year by Baltimore Magazine for my work in safeguarding the election process;
- In 2004, I received the Electronic Frontiers Foundation Pioneer Award; and
- In 2000, I received the Best Paper Award at the 9th USENIX Security Symposium.

17. I serve and have served on several technical and editorial advisory boards. For example, I served on the Editorial and Advisory Board for the International Journal of Information and Computer Security. I also served on the Editorial Board for the Journal of Privacy Technology. In addition, I have been Associate Editor of the IEEE Security and Privacy Magazine and served as Associate Editor of the Association for Computing Machinery's ("ACM") Transactions on Internet Technology. I also served as Associate Editor of the journal Communications of the ACM, and I was an Advisory Board Member of Springer's Information Security and Cryptography Book Series. I also have served in the past as a member of the Defense Advanced Research Projects Agency's Information Science and Technology Study Group, a member of the Government Infosec Science

and Technology Study Group of Malicious Code, a member of the AT&T Intellectual Property Review Team, Associate Editor of the Electronic Commerce Research Journal, Co-editor of the Electronic Newsletter of the IEEE Technical Committee on Security and Privacy, a member of the board of directors of the USENIX Association (the leading academic computing systems society), and a member of the editorial board of the Bellcore Security Update Newsletter.

18. I have spoken on information security and electronic privacy issues at more than 50 seminars and symposia. For example, I presented keynote addresses on the topics "Security of Electronic Voting" to Computer Security 2004 Mexico in Mexico City, Mexico, in May 2004; "Electronic Voting" to the Secure Trusted Systems Consortium 5th Annual Symposium in Washington, D.C., in December 2003; "Security Problems on the Web" to the AT&T EUA Customer Conference in March 2000; and "Security on the Internet" to the AT&T Security Workshop in June 1997. I also presented a talk about hacking devices at the TEDx conference in October 2011 and another TEDx talk on the same topic in September 2015.

19. I am a named inventor on 10 United States patents, including U.S. Patent Nos. 5,638,446 ("Method for the Secure Distribution of Electronic Files in a Distributed Environment") and 5,809,140 ("Session Key Distribution Using Smart Cards").

20. A detailed record of my professional qualifications is set forth in the attached Exhibit 2002, which is my curriculum vitae, including a list of publications, awards, research grants, and professional activities. My curriculum vitae also lists the matters in which I have served as an expert.

II. UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES

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

22. I understand that a patent claim is invalid if it is anticipated or obvious in view of the prior art, and that a claim can be unpatentable even if all of the requirements of the claim cannot be found in a single prior art reference. I further understand that invalidity of a claim requires that the claim be anticipated or obvious from the perspective of a person of ordinary skill in the art at the time the invention was made.

23. I have been informed that a patent claim is invalid if it would have been obvious to a person of ordinary skill in the art. In analyzing the obviousness of a claim, I understand the following factors may be taken into account: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims;

(3) the level of ordinary skill in the art; and (4) any so called "secondary considerations" of nonobviousness if they are present.

24. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. I understand that to prove that prior art or a combination of prior art renders a patent obvious, it is necessary to:

1) Identify the particular references that, singly or in combination, make the patent obvious;
2) Specifically identify which elements of the patent claim appear in each of the asserted references; and
3) Explain why a person of ordinary skill in the art would have combined the references, and how they would have done so, to create the inventions claimed in the patent. I further understand that exemplary rationales that may support a conclusion of obviousness include:
   - Combining prior art elements according to known methods to yield predictable results;
   - Simple substitution of known element for another to obtain predictable results;
   - Use of known technique(s) 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 the work for use in either the same field or a different field based on design incentives or other market forces if the variations are predictable to a person of ordinary skill in the art; and
- Some teaching, suggestion, or motivation in the prior art that would have led a person of ordinary skill in the art to modify the prior art reference or to combine the prior art reference teachings to arrive at the claimed invention.

25. I have been informed that, in considering obviousness, hindsight reasoning derived from the patent-at-issue may not be used.

III. STATE OF THE PRIOR ART

A. Internet Communication Channels

26. Internet communication channels were primarily focused on facilitating the transmission of data over the internet between different devices (like client devices, servers, and network infrastructures). By 2016, the framework for internet communications was an established idea, with the Open System Interconnection (OSI) model serving as the foundational structure. The OSI model is a conceptual framework used to understand and standardize how different networking protocols

interact to enable communication between computers and devices over a network, developed by the International Organization for Standardization in 1984.[1]

27. The OSI model divides network communication into seven layers, each responsible for specific tasks in the data transmission process. With these layers, networks can be broken up to manageable pieces. Starting at Layer 1 with the physical hardware, and moving up to Layer 7 with software applications, the OSI model shows how network communication spans from hardware raw data transmission to user-facing services in software. These OSI layers are:[2]

- **Physical Layer (Layer 1)**: Handles the transmission of raw bits over a physical medium (e.g., cables, radio waves).
- **Data Link Layer (Layer 2)**: Provides error detection and correction and manages data frames between devices on the same network (e.g., Ethernet, Wi-Fi).
- **Network Layer (Layer 3)**: Responsible for routing data packets across different networks using logical addressing (e.g., IP addresses, discussed below). This layer manages how data moves from the source to the destination across multiple hops.

---

[1] The OSI Model: An Overview. https://sansorg.egnyte.com/dl/oyHUTHkIzn.

[2] *Id.*

- **Transport Layer (Layer 4)**: Ensures reliable data transfer between systems, providing error recovery, flow control, and data segmentation (e.g., TCP, UDP, discussed below).
- **Session Layer (Layer 5)**: Manages sessions or connections between applications, ensuring that data exchanges between systems are synchronized and properly maintained.
- **Presentation Layer (Layer 6)**: Translates data between the application and transport layers, dealing with data formats and encryption (discussed below).
- **Application Layer (Layer 7)**: The top layer, where network applications and services (like web browsers, video streaming applications, etc.) interact with the network. This layer provides the direct interface to users.

28. In the following subsections, I further discuss common concepts and protocols adopted in the OSI model.

1. ***Network Layer (Layer 3): Internet Protocol Addresses***

29. The Network Layer is a key part of how devices communicate over the internet, and it uses Internet Protocol (IP) addresses to help route data between devices. Each IP address is a unique identifier assigned to each device connected to a network.

30. IP addresses are used by networking devices (like routers) to route data across the network – IP addresses help networking devices identify both the source and destination of the data. When a sender sends data, the IP address is used to determine the destination network, and routers along the way use it to forward the data toward the correct destination. Routers examine the destination IP address in each data packet to determine the best path for that packet to take, often using a process called routing, which can involve multiple hops through different networks.

31. There are two primary versions of IP addresses used today: IPv4 and IPv6. IPv4 addresses are 32-bit numbers, typically written in dotted-decimal notation (e.g., 74.125.227.160), while IPv6 addresses are 128-bit and written in hexadecimal format (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334), which allows for a vastly larger address space due to the increasing number of connected devices.

32. Human beings also utilize domain names (human-readable addresses like www.example.com) to identify and access resources on the internet. Each resource, e.g., a networked device with data for accessing, has its own IP addresses for communication. Therefore, the Domain Name System (DNS) is utilized to translate human-readable domain names (such as www.example.com) into machine-readable IP addresses (like 74.125.227.160). The DNS deploys DNS servers that maintain a database of domain names and their associated IP addresses for the translation, which mostly operates at the Application Layer (Layer 7).

2. *Transport Layer (Layer 4): Ports and protocols like TCP, UDP*

33. In the Transport Layer, "ports" are used to identify specific services or applications running on a device. Ports are numerical identifiers used by transport layer protocols (like TCP and UDP) to direct network traffic to specific applications or services running on a device. A device can have many applications or services running at once (like a web server, email server, or file transfer service) at the same time. Each service is assigned a unique port number, so that the correct data is sent to the correct service.

34. A port is a 16-bit number, ranging from 0 to 65535, assigned to a specific application or service. Of all ports, "Well-Known Ports" (Ports 0-1023) are reserved for widely-used protocols and services, e.g., Port 80 – HTTP (web traffic), Port 443 – HTTPS (secure web traffic), Port 25 – SMTP (email). The rest of the ports are "Registered Ports" (Ports 1024-49151) assigned by the Internet Assigned Numbers Authority (IANA) to specific applications and services, and Dynamic/Private Ports (Ports 49152-65535) used for temporary or private communication between devices or applications, often assigned dynamically during runtime.

35. Transmission Control Protocol (TCP) is a connection-oriented protocol commonly used in the Transport Layer. It establishes a connection before transmitting data and guarantees that data packets are delivered in the correct order,

without duplication, and with error correction. TCP is commonly used for applications where data integrity and reliability are critical, such as web browsing (HTTP/HTTPS) and email (SMTP).

36. User Datagram Protocol (UDP) is a connectionless protocol in the Transport Layer that is also commonly used in the Transport Layer. UDP offers faster data transmission by avoiding the overhead of establishing a connection or ensuring reliability. Unlike TCP, UDP does not guarantee delivery, order, or error correction.

37. TCP and UDP both utilize port numbers to identify specific services or applications running on a device. When data is transmitted over a network, TCP and UDP rely on port numbers to ensure that the data is sent to the correct process or application on the receiving device.

38. In computer networking, a "socket" represents one end of a two-way communication channel. i.e., a socket identifies a sender's service that sends the communication, or a receiver's service that receives the communication. A socket relies on the transport layer protocols, such as TCP and UDP. A socket may, in some cases, be identified by an IP address and a port number, as discussed below. However, sockets need not always be identified by an IP address and a port number. For example, "raw sockets" are well-known in the art to be identified by protocol, rather than by port number and IP address. *See* EX-2019,

https://www.opensourceforu.com/2015/03/a-guide-to-using-raw-sockets/ ("There is no need to provide the port and IP address to a raw socket, unlike in the case of stream and datagram sockets."); EX-2020, https://learn.microsoft.com/en-us/windows/win32/winsock/tcp-ip-raw-sockets-2.

3. ***Data transmission with IP Addresses and Ports***

39. As discussed above, IP addresses and ports are fundamental elements that facilitate data transmissions across the internet. The following is an example of data transmission with an IP address and a port number in the '697 patent:

Connected to we109.media.castr.live (198.204.252.202) port 443

(EX-1006, 36:18-54.)

40. This demonstrates a network connection to a web server we109.media.castr.live:

- **we109.media.castr.live:** This is the hostname or domain name of the web server that the user tries to connect to.
- **98.204.252.202:** This is the IP address of the web server (Network Layer).
- **Port 443:** This specifies the port number on which the web server is listening for incoming connections (Transport Layer). Port 443 is commonly used for HTTPS (secure HTTP) traffic. This means that the server is likely hosting a secure web service.

41. Here, a transmission "socket" is identified on the web server that receives this communication. The socket is identified with the IP address (198.204.252.202) and port number (443) in the above connection.

42. Data transmission in this network connection (the "client-server" model) is discussed in the subsection below.

4. ***The Client-Server Model in Network Transmissions***

43. The "client-server model" in network communications is a framework where one device, known as the "client," requests services or resources, and another device, known as the "server," provides them. The client-server model is an established idea by 2016. [3]

44. In this model, clients are typically devices or applications that make requests over a network, such as a web browser requesting a webpage by a user. Servers are typically systems that respond to these requests, often hosting services like websites, databases, or files.

45. The client-server interaction is typically structured as a "request-response" cycle, where the client sends a request (e.g., an HTTP request) to the

[3] *See,* e.g., EX-2021, Oracle documentation in 2011: Anatomy of the Client/Server Model. https://docs.oracle.com/cd/E13203_01/tuxedo/tux80/atmi/intbas3.htm.

server, and the server processes the request and sends back the appropriate response (e.g., the requested web page).

46. The following is an example of data transmission utilizing the client-server model in the ’697 patent:

Connected to we109.media.castr.live (198.204.252.202) port 443

(EX-1006, 36:18-54.)

47. This demonstrates the following client-server communication:

- **Client:** The client is the device trying to access resources (e.g., a video stream) hosted by the server at we109.media.castr.live. The client sends a request to the server to access or interact with the service provided by the server. In this case, the client is the “viewer client device” specified in the ’697 patent. (EX-1006, 35:9-36:14.)
- **Server:** The server is the web server identified by the domain name we109.media.castr.live (which resolves to the IP address 198.204.252.202). The server listens for incoming connections on port 443, which is the standard port for HTTPS traffic.
- **Request-Response:** The client sends a request to the server (“Connected to...”), which is received by the server at port 443. In the client-server model, the server processes this request from the client – such as

retrieving streaming media – and then sends a response back to the client.

- **TCP Connection:** The use of port 443 indicates that the client and server are communicating securely over TCP.

48. In a client-server model, a client would communicate with a server to request for specific data. On the other hand, networked communications, including Internet communications, also allow for other types of communication models. As an example, peer-to-peer (P2P) is a decentralized network architecture where devices (peers) communicate directly with each other, sharing resources or data without relying on a server.

5. ***Content Delivery Networks (CDNs)***

49. A Content Delivery Network (CDN) is a system of distributed servers that work together to deliver web content, media files, and other online resources to users based on their geographic location. (*See, e.g.*, EX-2022, Peng, Gang. "CDN: Content distribution network." *arXiv preprint cs/0411069* (2004) https://arxiv.org/pdf/cs/0411069.)

50. In a user's interaction with a web resource, "latency" plays a crucial role in determining the overall user experience. Latency is the time it takes for data to be generated (i.e., by the server) to be consumed (i.e., viewed by the client). Geographically closer servers are a way to improve latency because the data travels

a shorter physical distance, reducing the time it takes for the request and response to reach their destination.

51. Therefore, a CDN deploys multiple geographically dispersed servers (called Points of Presence, PoPs) to improve website performance by reducing latency and load times. When a user requests content, the CDN directs the request to the closest server in its network, minimizing the distance data needs to travel. This not only speeds up content delivery but also helps with scalability, as CDNs can handle large amounts of traffic by distributing the load across multiple servers. Additionally, CDNs often provide benefits like redundancy (reducing the risk of downtime) and security (if one server is unavailable due to a cyberattack, other servers can still provide web content to clients).

52. A CDN still operates within the client-server model by maintaining the basic structure of clients requesting resources from servers, but it optimizes this interaction by distributing content across a network of geographically dispersed servers.

B. Real Time Content Sharing Protocols

53. By 2016, real-time content sharing technologies are being integrated into social media platforms and video-sharing platforms. "Streaming" refers to the process of delivering video and audio content from a remote server to a user's device in real-time. (EX-2023, The History of Streaming Media.

https://boyetblog.s3.amazonaws.com/PCPlus/324.Streaming.pdf.) The audio and/or video content shared in streaming is generated by a broadcaster and is delivered over the Internet to viewers for concurrent playback on their devices. The broadcaster's live stream can include video-based commentary about an event, along with real-time event data, such as scores or other relevant information. This streaming process allows viewers to consume the content with minimal latency, ensuring a near-instantaneous viewing experience, often enhanced by interactive elements like chat, event updates, and special graphics, to engage the audience on their local devices.

54. The following subsections discuss streaming protocols utilized in the real-time content sharing, including streaming services.

1. ***Persistent/Continuous Streaming Transmission Protocols***

55. Persistent streaming protocols focus on real-time, interactive delivery of multimedia data.

a. *Real Time Streaming Protocol (RTSP)*

56. The Real-Time Streaming Protocol (RTSP) is a network protocol designed for streaming media over IP networks. It facilitates the control of multimedia content on a server, allowing clients to "PLAY" and "PAUSE" with a real-time interaction to the media stream. RTSP was introduced by the Internet Engineering Task Force (IETF) as a standard protocol for controlling the delivery of multimedia content over IP networks in RFC 2326, published in April 1998. (EX-

2024, RFC 2326: Real Time Streaming Protocol (RTSP) https://www.ietf.org/rfc/rfc2326.txt.)

b. *Real Time Messaging Protocol (RTMP)*

57. The Real-Time Messaging Protocol (RTMP) is a communication protocol for the delivery of audio, video, and data over the internet, particularly for streaming applications. The RTMP was originally designed for streaming to Flash Player, a once popular software utilized to view multimedia content. On December 21, 2012, Adobe received RTMP Specification version 1.0. (EX-2025, Adobe's Real Time Messaging Protocol https://rtmp.veriskope.com/pdf/rtmp_specification_1.0.pdf.)

c. *Web Real-Time Communication (WebRTC)*

58. The Web Real-Time Communication (WebRTC) Protocol is an open-source technology that allows users to exchange audio, video, and data in real time in their web browsers, without requiring additional plugins or third-party software. WebRTC also assists audio and video sharing in a P2P communication model. WebRTC was first introduced in 2011 by Google as an open standard and has since been developed by the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF). (EX-2026, WebRTC 1.0: Real-time Communication Between Browsers https://www.w3.org/TR/2011/WD-webrtc-20111027/.)

2. *Adaptive Bitrate (ABR) Protocols*

59. While persistent streaming protocols are more suited for real-time, interactive content delivery, Adaptive Bitrate (ABR) protocols are designed to optimize the quality of streaming media based on varying network conditions.

a. *Apple's HTTP Live Streaming (HLS) Protocol.*

60. HTTP Live Streaming (HLS) is a media streaming protocol first introduced by Apple in 2009. (EX-2027, HTTP Live Streaming: draft-pantos-http-live-streaming-01 https://datatracker.ietf.org/doc/html/draft-pantos-http-live-streaming-01.) HLS is intended to deliver audio and video content over HTTP by breaking the media into small chunks (typically 10 seconds each) and serving them through standard web servers. HLS is adaptive, meaning it can adjust the quality of the stream in real-time based on the viewer's internet speed, ensuring smooth playback even in fluctuating network conditions.

61. One downside of HLS is its relatively high latency compared to other real-time protocols discussed above, making it less ideal for real-time applications. A key factor is that HLS renders media into small chunks, then serves these chunks over HTTP as individual files (e.g., .ts files for video segments). This requires the client to download a complete chunk before playing it, which adds on to the latency. In addition, as I discussed above with regard to Transport Layers in the OSI Model,

HTTP connections run on TCP, which prioritizes data integrity and reliability instead of prioritizing on reducing latency.

b. *Microsoft's HTTP Smooth Streaming (MSS) Protocol*

62. Microsoft's HTTP Smooth Streaming (MSS) is an adaptive bitrate streaming protocol designed for delivering high-quality video and audio content over HTTP. Like HLS, MSS breaks media into small segments and dynamically adjusts the quality of the stream based on the viewer's network conditions. MSS was introduced by Microsoft in 2008 as part of its Silverlight platform to provide adaptive streaming solutions for Windows-based devices and services. (EX-2028, Microsoft To Add Smooth Streaming in IIS7 https://virtualizationreview.com/articles/2008/11/04/microsoft-to-add-smooth-streaming-in-iis7.aspx.)

63. MSS faces the same challenges with regard to latency, compared to HLS.

c. *Adobe's HTTP Dynamic Streaming (HDS) Protocol*

64. Adobe's HTTP Dynamic Streaming (HDS) similarly works by breaking the media into small, HTTP-served chunks and adjusting the quality of the video stream based on the viewer's available bandwidth. It is optimized for use with Adobe Flash Player. HDS was introduced by Adobe in 2010 as a solution for streaming content to Flash-enabled devices. (EX-2029, HTTP Dynamic Streaming. (Archived May 14, 2010)

https://web.archive.org/web/20100514170442/http://www.adobe.com/products/httpdynamicstreaming/.)

65. HDS faces the same challenges with regard to latency, compared to HLS and MSS.

d. *Standards-based ABR protocols, e.g. MPEG-DASH.*

66. During the late 2000s and early 2010s, multiple open standards are also established for additional implementation of ABR protocols. The Dynamic Adaptive Streaming over HTTP (MPEG-DASH) is an example. MPEG-DASH is an international standard for ABR streaming, initially developed by the Moving Picture Experts Group (MPEG) in 2011. (EX-2030, MPEG ratifies its draft standard for DASH. (Archived December 14, 2011) https://web.archive.org/web/20111214104626/http://mpeg.chiariglione.org/meetings/geneva11-1/geneva_press.htm.) Similar to the ABR protocols discussed above, MPEG-DASH is also a generic and flexible ABR protocol that allows streaming of media over standard HTTP.

67. Standards-based ABR Protocols also face the same challenges with regard to latency, compared to HLS, MSS, and HDS.

C. Live Audio/Video Streaming Services in 2016

68. Live audio and video streaming technology in 2016 was rapidly evolving, driven by the increasing demand for live content on platforms like Livestream, Twitch.tv, YouTube Live, Facebook Live, and Musical.ly.

69. Livestream (livestream.com) was initially founded as Mogulus in 2007, before rebranding as Livestream in 2009. (EX-2031, Mogulus Rebrands With A Killer Domain: Livestream.com https://techcrunch.com/2009/05/18/mogulus-gets-a-killer-domain-for-rebranding-livestreamcom/.) Livestream (Mogulus)'s livestreaming functionalities not only enabled individual broadcasters, but also assisted in news broadcasting, such as livestreaming the U.S. president inauguration in 2009 in collaboration with C-SPAN: "C-SPAN will debut its Inauguration Hub on January 20, featuring an online 'control room'--a multichannel grid designed by Mogulus with Webcasts of inauguration activities. Visitors will be able to choose from one of four live feeds featuring events like the swearing in at the Capitol, the parade, and a number of inaugural balls." (EX-2032, How to watch Obama's inauguration (and the parties) online https://www.cnet.com/tech/tech-industry/how-to-watch-obamas-inauguration-and-the-parties-online/.) This indicates that visitors to C-SPAN's "Inauguration Hub" website will have the option to select from four different live video streams, each showing a different channel of the inauguration events. For example, one video stream might show the swearing-in ceremony at the

Capitol, another might show the parade, while others might cover different inaugural balls or other related activities. The viewers can pick which individual live stream they want to watch, allowing them to focus on the particular event they're most interested in.

70. In 2016, Twitch.tv was the dominant platform for live-streaming, with a focus on streaming games and interactive content, combined with integrated chat features allowing for a more community-driven experience. In 2014, Twitch was already marked as 4th in peak U.S. Internet traffic. (EX-2033, Twitch is 4th in Peak US Internet Traffic https://blog.twitch.tv/en/2014/02/05/twitch-is-4th-in-peak-us-internet-traffic-90b1295af358/.)

71. Twitch's infrastructure is designed to support a global, real-time, high-quality live video streaming platform with a large number (i.e., millions) of concurrent users. According to Twitch documentation in December 2015 about its infrastructure, at its core, Twitch utilizes a "video system" that handles the ingestion of live streams from broadcasters via RTMP, then transcodes them into multiple formats (i.e., HLS) for efficient delivery to viewers. The video streams are then distributed across geographically dispersed PoPs. (EX-2034, Twitch Engineering: An Introduction and Overview https://blog.twitch.tv/en/2015/12/18/twitch-engineering-an-introduction-and-overview-a23917b71a25/.) As I discussed above

with regard to CDN, geographically dispersed PoPs reduce the distance of data transmission, which reduces low-latency delivery to users.

72. However, Live video streaming often had noticeable latency in 2016. Even with optimization like the use of PoPs in CDNs, there was still noticeable latencies in various processes live streaming services. For example, a noticeable broadcasting latency presents between when broadcasting content was generated by a broadcaster, and when the livestream was viewed by a viewer. Latency negatively affects real-time events like sports or news broadcasts, which was a major focus for improvement.

73. Broadcasting latency was recognized by stream platforms around mid-2010s. For example, an academic article in 2015 defines “broadcast latency” as “the time lag of a live event when viewers watch the live streaming from the source,” reflects “a viewer’s time difference with the commentator and other viewers when they watch and discuss the current live event.” The article recognizes that “there is significant disparity between the long broadcast latency and the short live messaging latency” in Twitch services. In various tests conducted by the researchers, the broadcast latency ranged from 12 to 21 seconds, meaning that a viewer watching a live stream on Twitch would experience the content 12 to 21 seconds after someone attending the live event in person. (EX-2035, Zhang, Cong, and Jiangchuan Liu. "On crowdsourced interactive live streaming: a twitch. tv-based measurement study."

In *Proceedings of the 25th ACM workshop on network and operating systems support for digital audio and video*, pp. 55-60, 2015 https://arxiv.org/pdf/1502.04666.) A 12 to 21-second delay in sporting events can be significant because it creates a time gap where viewers may learn key outcomes (like goals, plays, or injuries in a sporting event) before they see it on the stream. This broadcasting delay greatly impact the viewer experience, especially for fans engaged in real-time reactions, live betting, or those participating in discussions on social media or chat platforms. As recognized by the article, "[a] long broadcast latency will obviously affect the interactivity" "the current delay strategy on the viewer's side substantially impacts the viewers' interactive experience." (*Id.*)

74. Issues related to the broadcasting delay in live streaming services were also recognized by the Patents-in-Suit. For example, the '697 patent discusses that "the live video streaming platform Twitch.tv has a latency of approximately 15 seconds, and Facebook's live streaming platform has an average latency of approximately 10 seconds." (EX-1006, 2:31-34.)

75. Broadcasting latencies probably could also be alleviated by duplicating the broadcasted content in multiple copies, each with a different streaming format, and giving users the opportunity to select different streaming formats provided by different media sources (e.g., different server components). Although different formats may lead to different quality of service (e.g., video resolution), users could

have the opportunity to choose a format that provides less resolution, but requires less data in transmission, hence less latency. Alternatively, if the user has better hardware and a stronger network connection, they can opt in for a streaming format that fits their needs (e.g., formats with better resolution or frame rate). (EX-1006, 2:34-47.) Twitch, as an example, only supplied users with one type of video streaming format in mid-2010s. According to Twitch's documentation in December 2015, Twitch's live stream multimedia content is received from broadcasters as RTMP only and is sent to viewers in HLS streams only: "We take the incoming RTMP stream from the broadcaster and transcode it into multiple HLS streams." (EX-2036, Twitch Engineering: An Introduction and Overview https://blog.twitch.tv/en/2015/12/18/twitch-engineering-an-introduction-and-overview-a23917b71a25/.)

76. In addition, live streaming platforms by 2016 did not implement a separation of communication channels for video information and other event information (such as scores, player stats, game updates, etc.). In 2016, platforms like LiveStream, Facebook, or Twitch did not appear to send event-based information to live stream viewers in a timely manner through a dedicated communication channel, which is separate from the video content itself. Event information was hard-embedded into the video signals and was delivered to viewers with the video stream. (EX-1006, 28:39-59 ("hard-embedded into livestreams of a sports broadcast").) Had

the data flows of video and event information been separated, delivering event-specific updates through dedicated channels likely could have allowed for more detailed and timely updates for viewers.

IV. THE '697 PATENT

A. Summary Of The '697 Patent

77. The '697 patent is entitled "Systems, apparatus, and methods for scalable low-latency viewing of broadcast digital content streams of live events, and synchronization of event information with viewed streams, via multiple internet channels" and describes and claims systems and processes related to video streaming with contextually matched event information. (*E.g.*, EX-1006, 4:41-65; 9:16-58; Claims 19, 23, 27.) The '697 patent describes "event information" as including, but not being limited to "one or more of team information (e.g., team names, abbreviations and/or logos), score information (e.g., with essentially real-time score updates synchronized with the video-based commentary), player information, venue information, game status information (e.g., on-base, at-bat, timeouts, fouls, pole position, yards-to-go, yards-to-goal, down), team statistics, player statistics and the like." (*E.g.*, EX-1006, 4:42-55.) The '697 patent further contemplates that such "event information" may yet further include "a wide variety of information germane to the event," any of which may be "displayed as event information (e.g., with essentially real-time updates of evolving information as the event progresses)

together with the rendering of the video-based commentary on respective client devices." (*E.g.*, EX-1006, 4:62-5:1.)

78. Unlike the systems and methods described in the prior art, which contend with a viewer-noticeable off-set between a delivered video stream and associated information (EX-1006, 4:30-34), or rely on a single channel that delivers both the video stream and the embedded associated information with significant latency (EX-1006, 28:39-59), the '697 patent describes and claims a novel and specific, technologically improved, one-to-many socket architecture. (EX-1006, 19:64-20:11; 21:37-52; 22:7-39.) I discuss this in further detail below.

79. The '697 patent describes a socket architecture that allows multiple viewer client devices to receive information from a dedicated event information channel specific to the event over an "event socket." (*E.g.*, EX-1006, 21:37-52; 22:7-39.) Figure 2 of the '697 patent depicts a socket server 600 and communicatively coupled client devices, such as viewer client device #1 (200A) and a viewer client device #3 (200C). (EX-1006, FIG. 2).

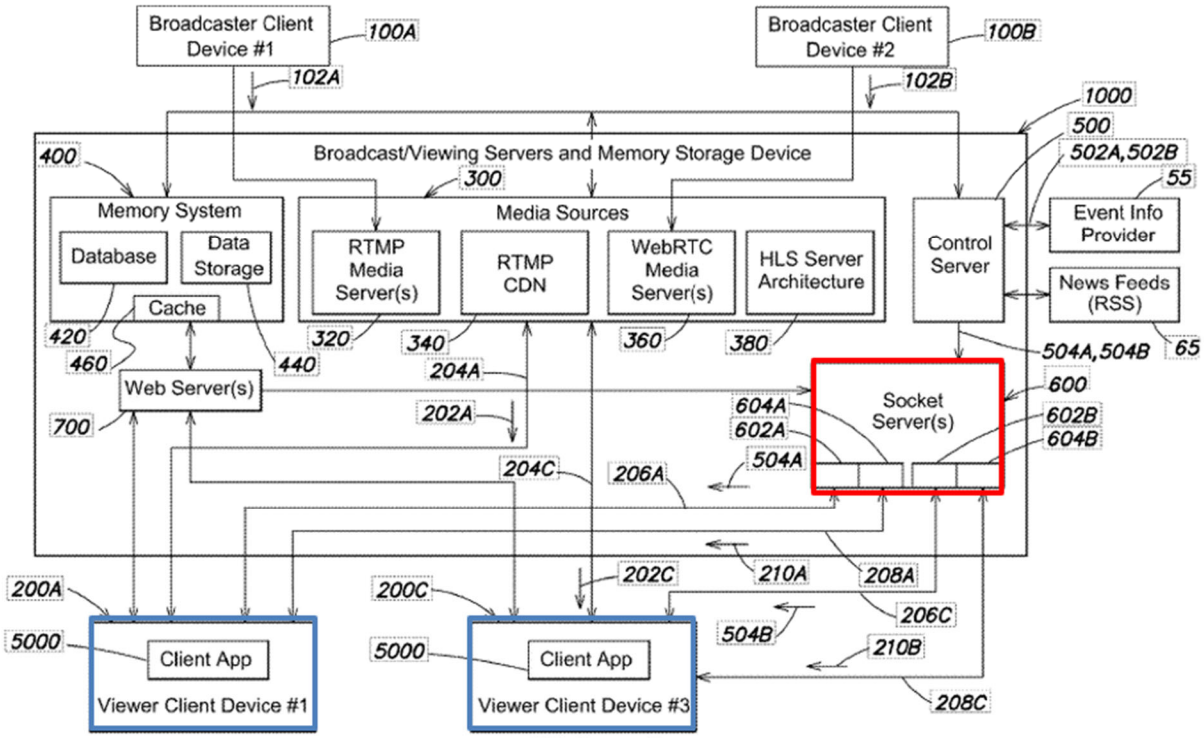


FIG. 2

(EX-1006, FIG. 2 (annotated).)

80. The ’697 patent describes the socket server(s) 600 as “facilitat[ing] communication, between a given broadcaster of a live stream of video-based commentary and corresponding viewers of copies of the live stream.” (EX-1006, 20:11-15.) To facilitate the foregoing, the socket server(s) 600 employ “sockets,” such as “one or more sockets of the socket server(s) 600 dedicated to the particular broadcaster’s live stream allow respective client devices used by the broadcaster and their viewers to establish a chat/system event channel with the socket server(s), such that chat messages/system event information is shared in a synchronized manner by

the broadcaster of the live stream and corresponding viewers of copies of the live stream." (EX-1006, 20:18-25.)

81. More specifically, the '697 patent describes overcoming the "technical challenges" of "provid[ing] the same event information, . . . in a synchronized and low-latency manner to all . . . broadcasters and viewers interested in following the same event." (EX-1006, 26:34-38.) The '697 patent proposes to address the foregoing challenge using a "socket-based 'event information channel' dedicated to the event" that allows "all viewers…of the event, regardless of which live stream they may be…watching," to receive the same event information. (EX-2003, 26:50-27:4.)

82. The '697 patent describes delivering such event information via "event sockets" of the socket server(s) 600. (*E.g.*, EX-1006, 21:37-52.) For example, the '697 patent describes using the socket server(s) 600 to "establish one or more ***first event sockets 602A*** dedicated to the first event information and one or more ***second event sockets 602B*** dedicated to the second event information." (*E.g.*, EX-1006, 21:49-52 (emphasis added).)

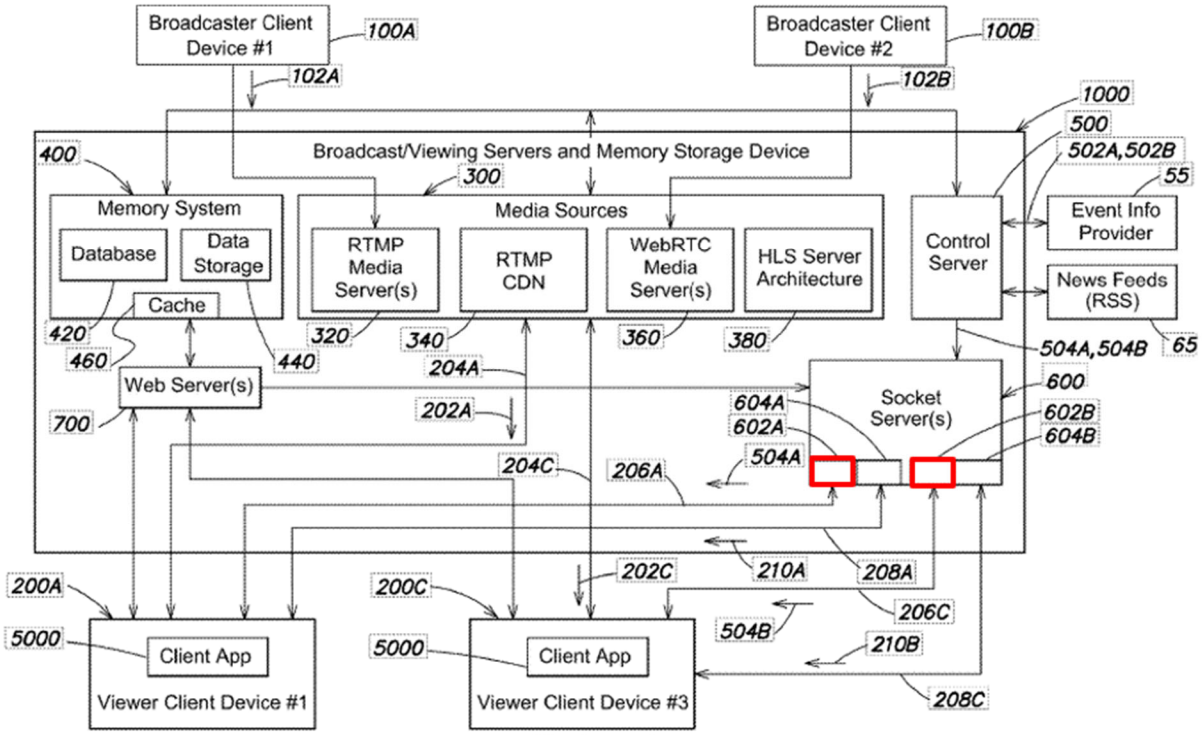


FIG. 2

(EX-1006, FIG. 2 (annotated).)

83. The event socket may support the synchronized delivery of real-time, contextually matched and actionable event information to each viewer client device that is contextually matched with a delivered video stream. (EX-1006, 21:37-52; 22:7-39.) For example, as described in the '697 patent, "all viewers (and broadcasters) of the event, regardless of which live stream [e.g., live video stream] they may be generating or watching, connect to one or more sockets of a socket server that is/are dedicated to the event." (EX-1006, 26:57-60.) Accordingly, regardless of which video stream a viewer receives, all such "live streams relating to the event are similarly synchronized to event information and updates to same."

(*E.g.*, EX-1006, 26:60-62.) This allows each viewer to receive the same event information, in a synchronized manner and updated in real-time, despite viewing the same event via one or more different video streams. (*E.g.*, EX-1006, 26:50-27:4.)

84. To facilitate the foregoing synchronization techniques, the ’697 patent describes transmitting the foregoing described “event information” to the client device separately from the video stream of the event associated with said event information. (*E.g.*, EX-1006, 21:37-52; 22:7-39; FIG. 2; Claims 19, 23, 27.) For example, the ’697 patent describes and claims transmitting such event information to the client device via an event information Internet communication channel (e.g., an event information communication channel 206A, 206C). (EX-1006, 9:16-58; 22:60-64.)

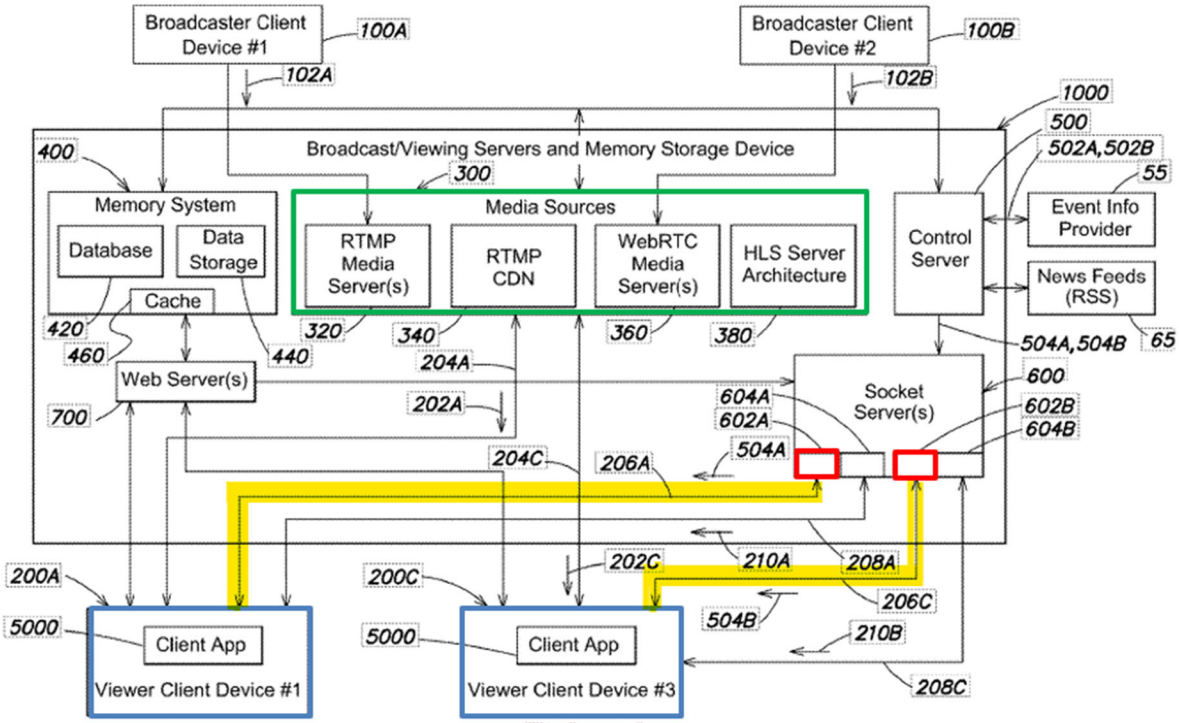


FIG. 2

85. The event information Internet communication channel is a separate channel from an associated video stream provided to the viewer device (e.g., video communication channel 204A, 204C). (EX-1006, 9:16-58; 22:57-60; 29:25-43; FIG. 2.)

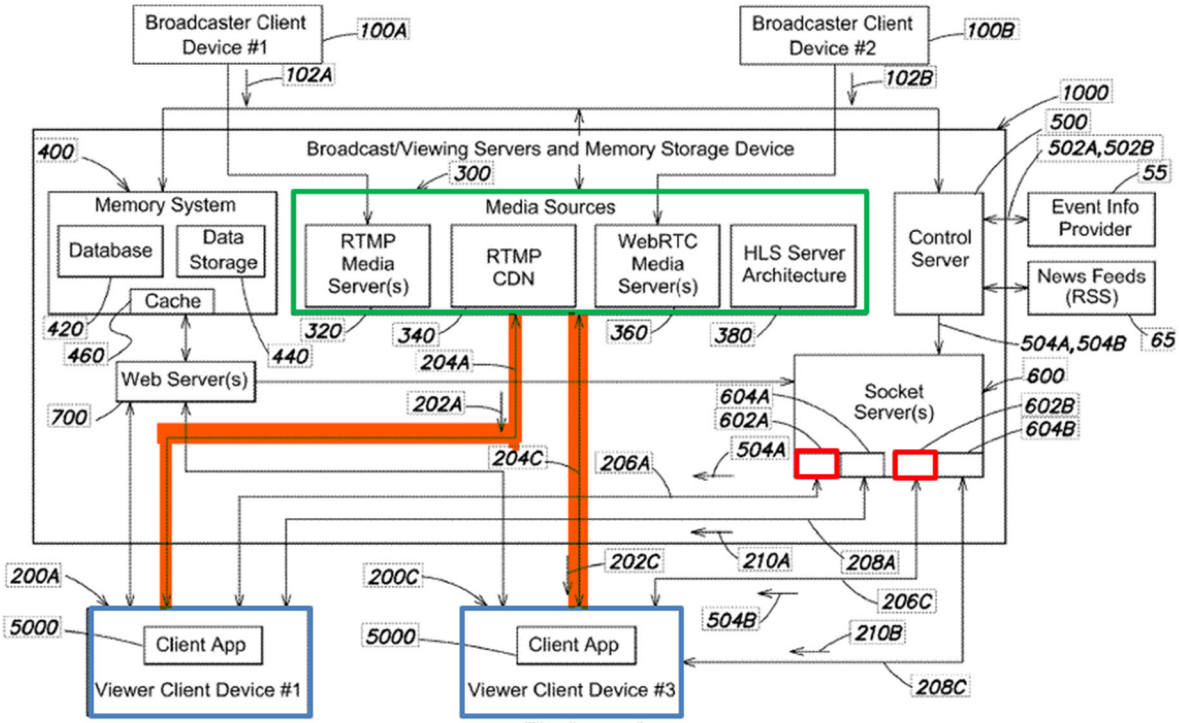


FIG. 2

(EX-1006, FIG. 2 (annotated).)

86. Among the innovations of the ’697 patent is a novel architecture that utilizes a dedicated event information channel to transmit event information, separate from the communication channel utilized for video streaming. (*E.g.*, EX-1006, 21:37-52; 27:5-27; FIG. 2.) By transmitting video signals to the viewer device separate from the event information, the ’697 patent facilitates the inclusion of real-time, contextually matched data alongside the video stream. (EX-1006, 9:16-58; 21:37-52; 27:5-27.)

87. For example, as described in the ’697 patent, “a variety of digital content format and transmission protocols are contemplated herein for the

broadcaster live streams." (EX-1006, 15:30-32.) Accordingly, the '697 patent describes video stream source selection appropriate for the "multiple live stream transmission formats, protocols, and access endpoints[, which] are contemplated for different types and numbers of viewers that may receive copies of broadcasters' live streams at different bitrates and with different qualities of service." (EX-1006, 27:39-44.) Specifically, "a viewer stream source selection algorithm which selects an appropriate media source for a given viewer" based on various properties, thereby resulting in different viewers receiving a video stream of the same event from different media sources. (EX-1006, 27:30-52.)

88. For example, as shown in FIG. 6, a plurality of media sources may be provided to facilitate the transmission of video content to a view client device. (EX-1006, 9:16-58.) In the example of FIG. 6, four possible media sources are shown: "1) the media server receiving the live stream in the first instance from a broadcaster (e.g., an RMTP media server 320 or a WebRTC media server 360); 2) an auto-scaling RTMP CDN server pool 340; or 3) an innovative HTTP Live Streaming (HLS) server architecture [380]." (EX-1006, 27:35-52; FIG. 6.) A coupled web server may "serve as a controllable switch" to couple a viewer client device to a respective media source based on a variety of factors, including, for example, to accommodate "viewer stream source selection algorithm which selects an appropriate media source for a given viewer based on, for example, the type of user

(e.g., VIP users, premium subscribers, media professionals) and the number of viewers of a particular broadcaster's live stream." (EX-1006, 27:35-52; 32:35-59; FIG. 6.)

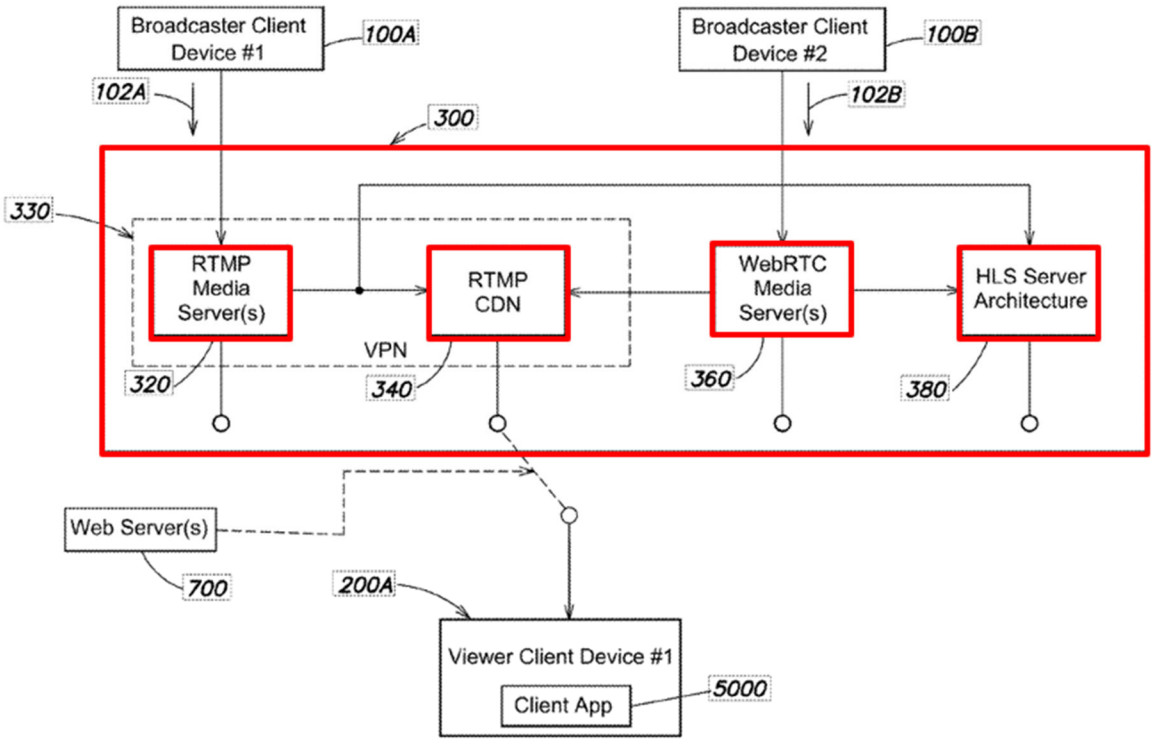


FIG. 6

(EX-1006, FIG. 6 (annotated).)

89. The separately provided event information supports all viewers of the same event (regardless of the video stream they receive) receiving contextually matched information for the event. (EX-1006, 9:16-58; 26:50-27:4; FIG. 2; FIG. 4.) For example, all viewers of event (regardless of the video stream they receive from the plurality of media sources) may receive synchronized information from the same event socket. (EX-1006, 9:16-58; 26:50-27:4; FIG. 2; FIG. 4.) In this regard, all live

streams of the event, across multiple viewer client devices, and across multiple different media sources, may therefore similarly receive synchronized event information, via the event information Internet communications channel. (EX-1006, 26:50-27:4.)

90. The ’697 patent demonstrates the flexibility provided by supplying the contextually matched event information on a separate event information channel. (EX-1006, 26:50-27:4.) For example, “if a viewer switches amongst different broadcasters of the same event (the viewer originally watches a first live stream from a first broadcaster of the event, and later selects a second live stream from a second broadcaster of the same event), the event information and updates to same (and any screen animations and special effects graphics that incorporate the event information) remain synchronized with all live streams from the different broadcasters, providing for a smooth second-screen experience across multiple broadcasters and viewers.” (EX-1006, 26:62-27:4.)

91. The ’697 patent describes systems that reduce and minimize latency by offering viewers the flexibility of receiving video streams according to various different streaming formats, which may lead to different quality of service. (EX-1006, 24:61-25:33; 26:50-27:4; 27:30-52.) The event data, as set forth in the ’697 patent, may be provided to all viewer client devices via an event socket, regardless of streaming format / quality of server, in support of providing synchronized data to

all such viewer devices. (EX-1006, 26:50-27:4.) In my opinion, the architecture disclosed in the ’697 patent to facilitate the foregoing (e.g., a dedicated event information channel to transmit event information from an event socket, separate from a communication channel utilized for video streaming) represents a significant improvement over the prior art.

B. Claim Construction

92. I understand that Petitioner contends the Board need not construe any claim terms aside from the means-plus-function limitations in this proceeding. (Petition, 6-8).

93. I further understand that in the co-pending district court litigation, Petitioner has proposed construction of nine claim terms recited in the ’697 patent. (EX-2003; EX-2004.)

1. “means for transmitting…”

94. Relevant to this proceeding, I understand Petitioner has construed the means-plus-function claim limitation 27.3 with the following construction:

Claim 27.3	**Petitioner’s Proposed Construction**	
	Structure	Function
“means for transmitting the first score information to at least the first viewer client	Socket Server performing the algorithm disclosed in 9:46-	Transmitting the first event information to at least the first viewer client device

Claim 27.3	Petitioner's Proposed Construction	
	Structure	Function
device via a first event information communication channel that is different from the first video communication channel."	58 and Figure 2 of the '697 patent.	via a first event information communication channel that is different from the first video communication channel.

95. For the purposes of this proceeding, I agree with Petitioner's construction; however, more specifically, a POSITA would understand the "means for transmitting" to be a socket server transmitting through an event socket that implements one or more of the algorithms described in Figures 2, 3, 21A-21E and accompanying disclosures, including 5:29-44 9:16-10:43, 19:64-20:61, 21:37-22:67, 23:20-45, 26:50-27:27, 29:25-43, 46:61-47:11, 51:27-49, and 53:21-37 of the '697 patent.

96. The '697 patent discloses a system having "at least one socket server (600)…to: receive from the control server at least the first score information; and transmit the first score information (504A) to at least the first viewer client device…via a first event information Internet communication channel (206A) between ***at least one first event socket*** (602A) of the at least one socket server and the first viewer client device." (EX-1006, 9:46-55, Fig. 2.)

97. The specification discloses algorithms that may be implemented by a socket server to perform the recited function. For example, the patent discloses a socket that distributes the same event information to different viewer client devices that are connected to the socket via the same EventID. *See*, *e.g.*, EX-1006, 23:20-45.)

98. The patent also discloses a socket that receives event information from a single point of entry and then distributes it to viewer client devices, resulting in a single synchronized update to all client devices and mitigating client-by-client latency. (*See, e.g.*, EX-1006, 26:50-27:27.)

99. The patent discloses similar examples a similar example at 46:61-47:11 (same).

100. The patent further discloses a socket that uses persistent authenticated connections to distribute event information with reference to portions of Fig. 21B related to socket connection. (*See, e.g.*, EX-1006, 51:27-49.)

101. Accordingly, a POSITA would understand "means for transmitting" to be a socket server transmitting event information through an event socket. I apply this understanding in my analysis, below.

C. Level Of Ordinary Skill In The Art

102. I understand that certain issues relating to the validity of the '697 patent must be judged from the perspective of a person of ordinary skill in the relevant art,

as I discuss below. I have been asked to define the level of a "person having ordinary skill in the art" or "POSITA" that someone would have had at the time the alleged invention as claimed was made. In deciding the level of ordinary skill, I have considered the following factors, which I have been informed are relevant in determining the level of ordinary skill in the art:

- The levels of education and experience of persons working in the field;
- The types of problems encountered in the art;
- The prior art solutions to these problems;
- The rapidity with which innovations are made; and
- The sophistication of the technology.

103. A POSITA at the time of the invention of the '697 patent (which claims priority to an earlier application (U.S. Provisional Application No. 62/371,558) filed August 5, 2016) would have had a bachelor's degree in electrical engineering, computer engineering, computer science, or equivalent degree and, in addition, approximately two years of work experience with Internet video streaming, Internet communication, and related protocols. More advanced degrees and/or training in a related discipline can compensate for shorter work experience.

104. I understand that Petitioner has defined the level of ordinary skill in the art as:

> A POSA in the field of computer networking 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 or an equivalent number of years of working experience, and one to two years of experience in networking environments, including at least some experience with live-streaming network architecture. Ex.1001 ¶¶72-76. Additional education or experience in related fields could compensate for deficits in the above qualifications. *Id*.

(Petition 13.)

105. Despite minor variations in these two positions, these variations would not alter my opinions expressed in this Declaration.

106. Based on either standard, I was at least a person of ordinary skill in the art at the time of the invention of the '697 patent.

V. OVERVIEW OF PRIOR ART

A. Overview Of Ellis

107. Ellis discloses systems and methods directed to an "intelligent tuning module" that facilitates a user (e.g., of a television) rapidly switching between programs of interest. (EX-1016, Abstract.) Unlike the '697 patent, Ellis does not disclose a socket server of any type or manner. Nor does Ellis concern the technical problem addressed by the '697 patent; namely, the synchronized delivery of real-

time, contextually matched and actionable event information across multiple viewer client devices with an associated video stream.

108. In discussing the deficiencies of "interactive media guidance applications," e.g., such as "interactive television program guide[s]," Ellis explains that "today's high-end user equipment devices ***may support multiple digital or analog tuners***." (EX-1016, ¶¶3-5 (emphasis added).) Ellis recognizes that "[c]urrent media guidance applications fail to ***efficiently manage and leverage the capabilities of multiple tuners*** to better match how certain types of viewers (e.g., sports viewers) commonly watch and interact with sporting events displayed on the user equipment device." (EX-1016, ¶5 (emphasis added).) This failure, according to Ellis, results in "idle tuners" that "are not optimized for certain types of viewing, such as sports viewing." EX-1016, ¶5.)

109. In the context of efficient management of ideal television tuners, Ellis discloses an "intelligent tuning module" that ultimately helps a user change a television channel. (EX-1016, Abstract; ¶5, 49, 137, 176, 178.)

110. The intelligent tuning module described in Ellis helps a user change a television channel, in part, by operating each tuner of a user equipment device (e.g., a television) to tune in, simultaneously, to a program of interest. (EX-1016, ¶¶56-59.) For example, a first tuner (e.g. tuner 202 as shown in Ellis FIG. 2A) may tune in to a first television program (e.g., a first sporting event) and a second tuner (e.g.,

tuner 204 as also shown in Ellis FIG. 2A) may tune in to a second television program (e.g., a second sporting event). (EX-1016, ¶¶5, 49, 57; FIG. 2A.) Ellis describes each of the example tuners as residing within the control circuity of example user equipment (e.g., a television). (EX-1016, ¶¶49-51.) Because the television is tuned to both (or more) events simultaneously, the television user may switch between watching either the first or the second sporting events quickly. (EX-1016, Abstract; ¶¶5, 137, 176, 178.)

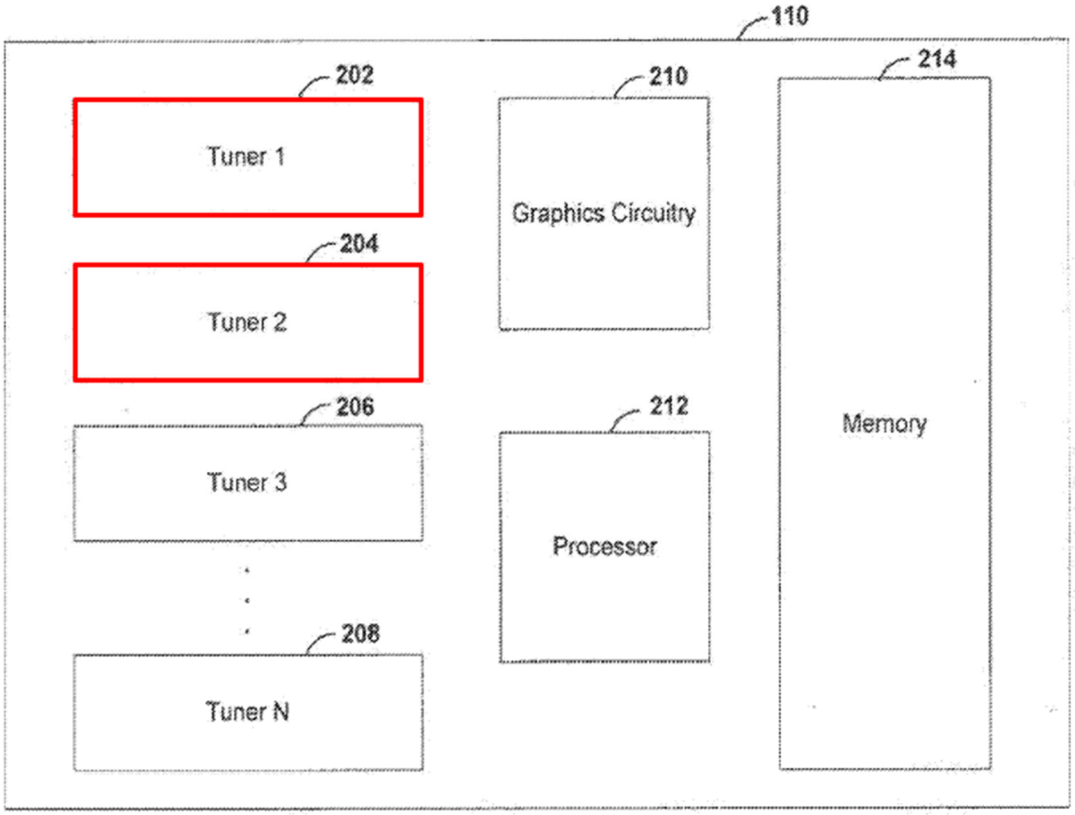


FIG. 2A

(EX-1016, FIG. 2A (annotated).)

111. Ellis proposes a "sports watcher application" to facilitate this "enhanced access and viewing functionality." (EX-1016, ¶8.) For example, Ellis describes the functionality to "designate a first sporting event as a primary sporting event and up to N−1 additional sporting events as secondary sporting events, where N is the number of available tuners in the user equipment device." (EX-1016, ¶8.) The interactive sports watcher application may, among other functions, facilitate the management of the tuners of the user equipment by: (i) ensuring that the primary sporting event always remains tuned, and (ii) that the secondary sporting events remain tuned, to the extent possible, based on the priority of said second sporting event. (EX-1016, ¶8.)

112. The designation of each respective event as primary / secondary, and the priority associated respective therewith may be managed by the interactive sports watcher application. (EX-1016, ¶¶8, 63, 64.) For example, the designation "may be based, at least in part, on user profile information, user preferences, active user monitoring, user history information, network viewing statistics, third-party ratings information (e.g., Nielson ratings), or any other suitable criterion or combination of criteria." (EX-1016, ¶8.)

113. Ellis proposes a solution to idle television tuners that—rather than tune each respective tuner passively to any program (or none at all)—tunes each respective tuner proactively based on the priority and ranking of content relevant to

the respective user of the television. (EX-1016, ¶¶64, 73, 164.) For example, Ellis describes an illustrative "tuner table" (e.g., tuner table 230 of FIG. 2B) in which each television content source (e.g., as represented by a "Source ID") is associated with a "Profile Score." (EX-1016, ¶¶62-64; FIG. 2B.) For example, the tuner table may include information indicative of "each sporting event available in the media system," which may be shown alongside a profile score that represents "a number of sports viewing preferences or criteria, such as the user's favorite sport or sports, favorite sports team or teams, favorite player or players, favorite venue or venues, favorite league or leagues, or any other suitable criteria, in the user's profile." (EX-1016, ¶¶62-64; FIG. 2B.)

220

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

230

Source ID (232)	Type (234)	Profile Score (236)
<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

FIG. 2B

(EX-1016, FIG. 2B (annotated).)

114. Ellis describes calculating the profile score using the sports watcher application, including by "[u]sing the weights assigned to each criterion and the rank assigned to each entry within each criterion." (EX-1016, ¶64.) Through this calculation, the Ellis profile scores having the highest value may be deemed of greatest interest to the user, and, therefore, can be used to determine a priority of the secondary television streams. (EX-1016, ¶64.) In this regard, the sports watcher application may itself be used to designate a given television stream as the primary or secondary games, for example, by selecting the program having the highest score as a primary, and lower ranking programs as secondary. (EX-1016, ¶64.)

115. In this manner, the programs having the highest score may be assigned to each of the available tuners. (EX-1016, ¶¶62-64.) For example, as shown in FIG. 2B, tuners (1) – (4) are each assigned to "tune" into the content having the highest profile score. Specifically, as depicted in FIG. 2B, Tuner 1 is shown as being tuned to "<sourceid_2>," which has a profile score of "95"; Tuner 2 is shown as being tuned to "<sourceid_1>," which has a profile score "90"; Tuner 3 is shown as being tuned to "<sourceid_8>," which has a profile score of "80"; and Tuner 4 is shown as being tuned to "<sourceid_4>, which has a profile score of "75." (EX-1016, ¶¶62-64; FIG. 2B.) The following content sources are not assigned to any of the Tuners (1) – (4) because each has a profile score that is less than the profile score of any of

the assigned content sources: "<sourceid_3>," "<sourceid_5>," "<sourceid_6>" and "<sourceid_7>." (EX-1016, ¶¶62-64; FIG. 2B.)

220

222 224

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

230

232 234 236

Source ID	Type	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

FIG. 2B

(EX-1016, FIG. 2B (annotated).)

116. Tuning into multiple streams simultaneously, according to Ellis, addresses the problem of "efficiently manag[ing] and leverag[ing] the capabilities of multiple tuners to better match how certain types of viewers (e.g., sports viewers) commonly watch and interact with sporting events displayed on the user equipment device." (EX-1016, ¶5.) For example, a sports television user may desire to view and/or be kept up to date generally with multiple sporting event simultaneously, and the disclosed interactive sports watcher application may address this desire, for

example, by providing information to the user and supporting rapid television channel changing (EX-1016, ¶¶56-57.)

117. To facilitate the foregoing, Ellis describes using the sports watcher application to enter a "sports watcher mode" in which an "interactive scoreboard" feature (e.g., interactive score board 302 as depicted in FIG. 3) may be displayed on a window of the user equipment. (EX-1016, ¶¶56-57; 68; FIG. 3.) The interactive scoreboard may be a graphical user interface that indicates various "passive and interactive features" to facilitate switching among television content streams tuned at each respective tuner. (EX-1016, ¶¶68, 69; FIG. 3.)

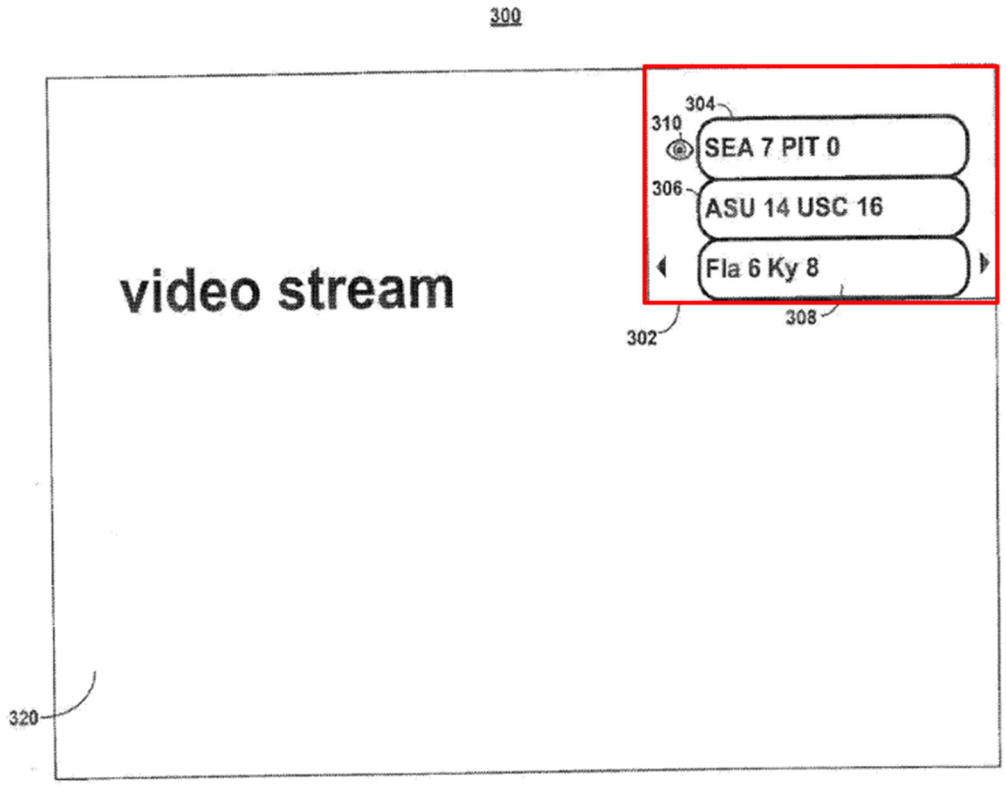


FIG. 3

(EX-1016, FIG. 3 (annotated).)

118. In connection with the interactive scoreboard, Ellis describes scoreboard entries that include information indicative of the "currently tuned sporting events." (EX-1016, ¶68.) For example, as described in Ellis, an "[i]con 310 may indicate that the game associated with the score listed next to icon 310 is viewable in an active or main window (e.g., in the foreground area of main window 320)." (EX-1016, ¶68; FIG. 3.) Further, "[i]con 310 may also indicate that the game associated with the score listed next to icon 310 is viewable somewhere on the current display screen (e.g., in a PIP or POP window)." (EX-1016, ¶68; FIG. 3.)

119. Further, the interactive score board may also facilitate the user changing the television channel, for example, by providing the user information that may prompt the user to change said channel. (EX-1016, ¶73; FIGS. 4, 5.) For example, the interactive scoreboard may include a television icon (FIG. 4), lighting bolt (FIG. 5), and/or other icon next to one or more scores listed in the interactive scoreboard. (EX-1016, ¶73; FIGS. 4, 5.) The television icon "may indicate that the game associated with the score is currently available on at least one television channel accessible by the user equipment device." (EX-1016, ¶73; FIG. 4.) In this regard, "television icon 404 (FIG. 4) provides more information to the user in a convenient, highly-visible way"; that is, the user can visually determine which games are

presently tuned in, and, if desired, can proceed to view said tuned games by selecting said television icon. (EX-1016, ¶73; FIG. 4.)

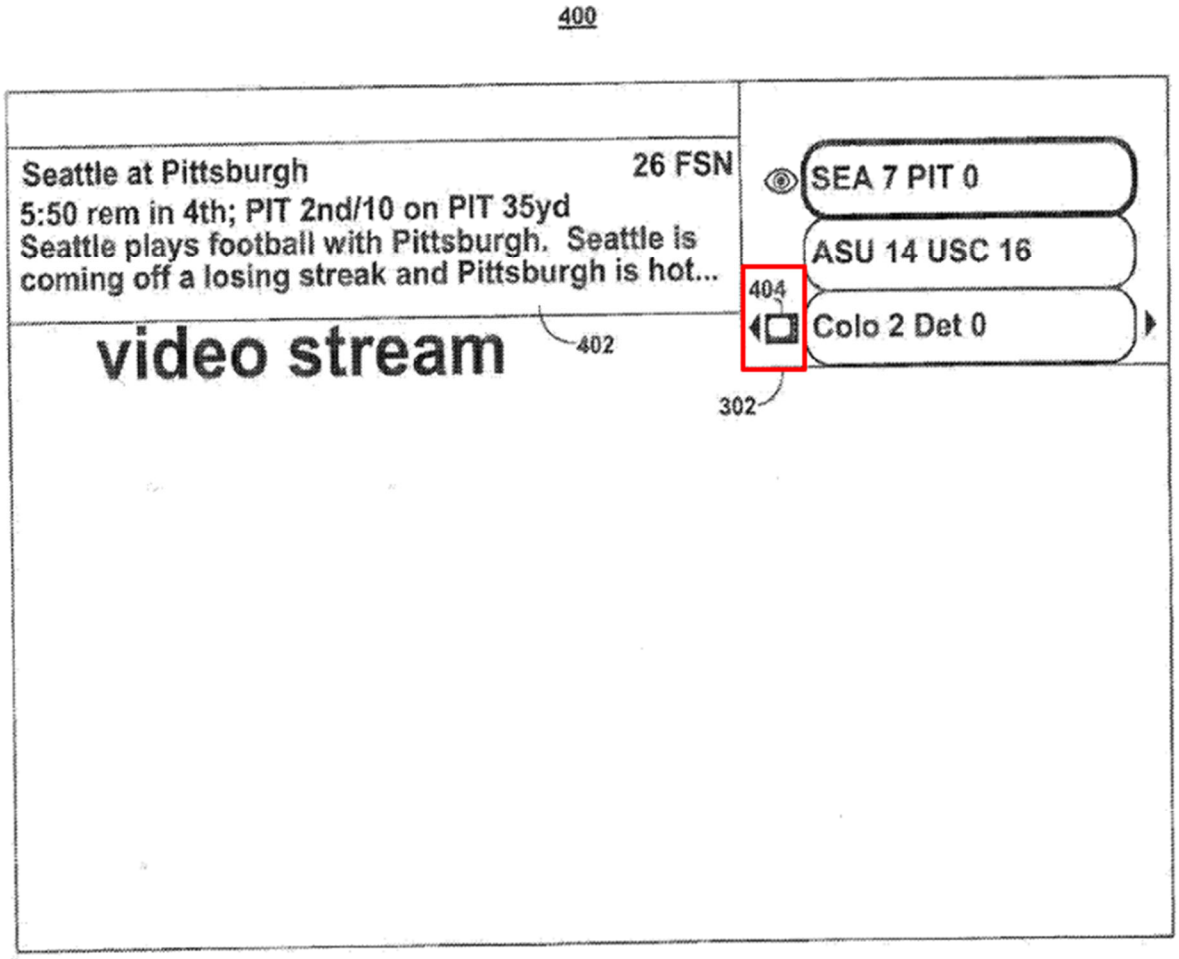


FIG. 4

(EX-1016, FIG. 4 (annotated).)

120. The scores provided in the interactive scoreboard may provide information to the television user of the "currently tuned sports events and games." (EX-1016, ¶68; FIG. 4.) Therefore, it is reasonable that a user may base his or her

decision to view one of the currently tuned sports events on the basis of said information; that is, a user may select the television icon based on the information (e.g., score) presented to the user about the tuned in, but not currently viewed sporting event. (EX-1016, ¶¶89-90; FIG. 5.)

121. To facilitate the foregoing functionality of the sports watcher application and the interactive scoreboard feature, Ellis describes a generic two-track architecture whereby the user's equipment "receives content in the form of signals from content source 130 and/or supplemental content source 120 over communications paths 150 and 152, respectively," for example, as shown in FIG. 1. (EX-1016, ¶39; FIG. 1.)

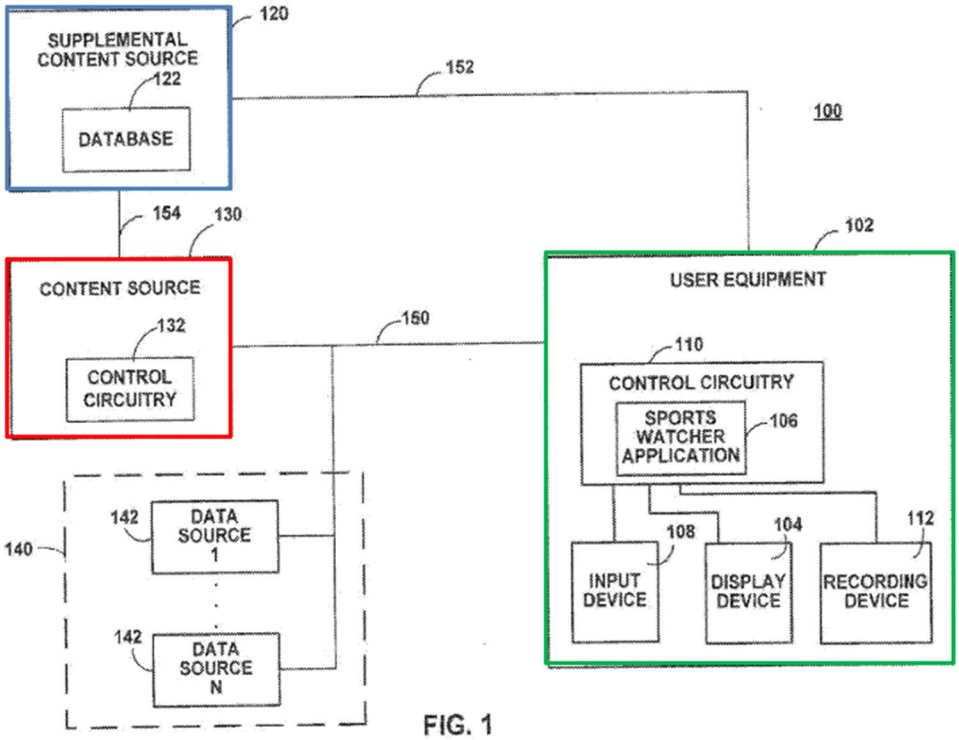


FIG. 1

(EX-1016, FIG. 1 (annotated).)

122. Ellis describes the content source 130 as a generic "source of media content," which may include "a cable system headend, satellite television distribution facility, television broadcast facility, on-demand server (e.g., video-on-demand (VOD) server), Internet or network media/web server, or any other suitable facility or system for originating or distributing passive or interactive content to user equipment 102." (EX-1016, ¶¶39, 40; FIG. 1.) Ellis goes on to state that generalized "media content" can be provided by the generic content source to the user equipment, including "broadcast programming, VOD programming, digital music, news, interactive applications (e.g., interactive games), Internet resources and web services (e.g., websites, newsgroups, and chat rooms), and any other content capable of being displayed." (EX-1016, ¶40.) The Ellis content source is therefore understood to provide, generally, video content to the user equipment (television). (*See generally* EX-1016.)

123. Absent from Ellis is any discussion of the particular protocols, systems, architecture and/or other techniques for delivery of video to the user equipment from the content source. (*See generally* EX-1016.) Ellis provides only a generic "communication path" for the provision of signals from the content source to the user equipment device, but again, the specific implementation of such path is notably absent from Ellis. (EX-1016, ¶45, FIG. 1.)

124. Ellis also describes the supplemental content source in generic terms. (EX-1016, ¶41.) According to Ellis, the supplemental content source "may be any suitable content source configured to provide real-time sports scores and game summary information, tournament brackets and tournament information, sports commentary, athlete biographies, spread and wagering data, and any other sports-related information to user equipment 102." (EX-1016, ¶41.) The Ellis supplemental content source is therefore understood to provide, generally, data to the user equipment (television). (*See generally* EX-1016.)

125. Absent from Ellis is any discussion of the particular protocols, systems, architecture and/or other techniques for delivery of data to the user equipment from the supplemental content source. (*See generally* EX-1016.) Indeed, I agree with the conclusion of Petitioner's expert, Dr. Wicker, in that "Ellis does not explicitly disclose the backend architecture/process flows for delivering the supplemental content data. (EX-1001, ¶106.) Ellis provides only a generic "communication path" for the provision of signals from the supplemental content source to the user equipment device, but again, the specific implementation of such path is notably absent from Ellis. (EX-1016, ¶45, FIG. 1.)

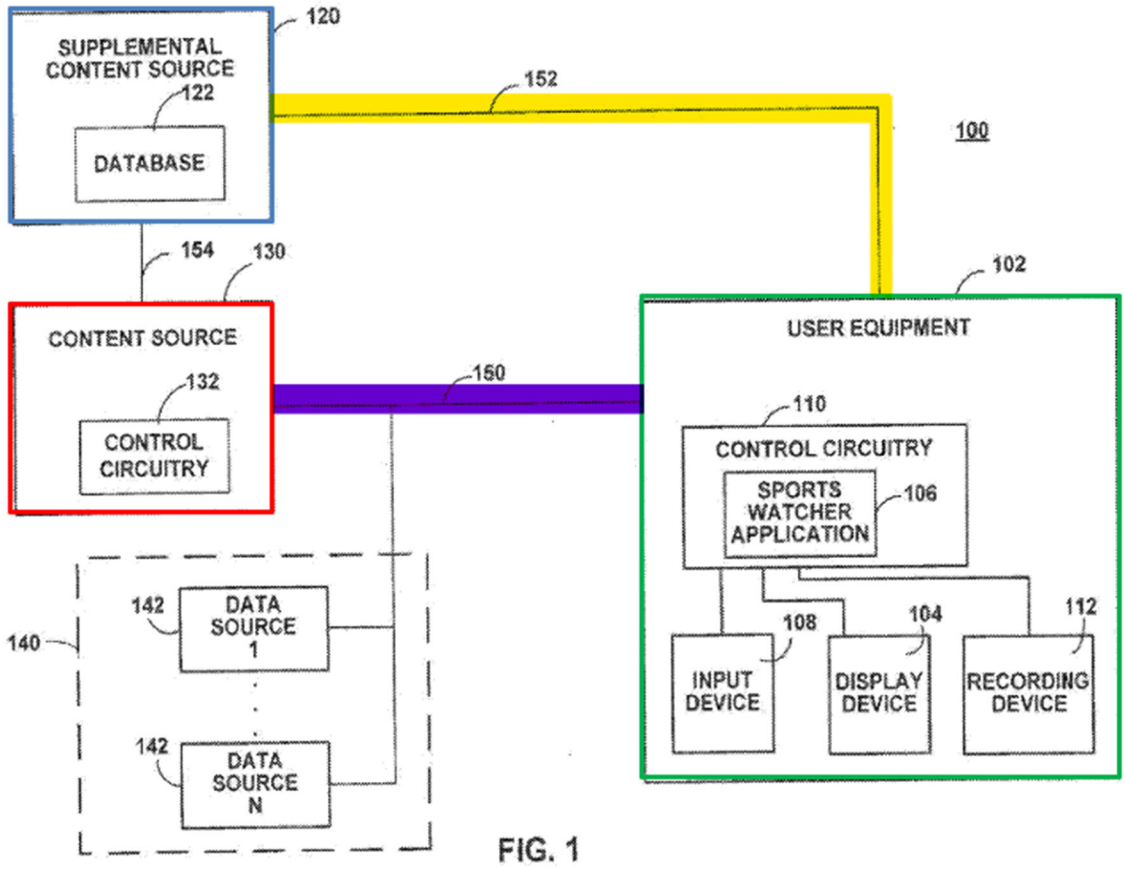


FIG. 1

(EX-1016, FIG. 1 (annotated).)

126. Ellis further discloses the presence of a database at the supplemental source that may store "all [] sports-related information," including "sports scores and game summary information." (EX-1016, ¶¶41-43.) In some cases, said database "may also store the interactive sports watcher application itself." (EX-1016, ¶44.) The Ellis database is therefore understood to store at least some (if not all) of the data and information transmitted to the user equipment by the supplement content source over the generic communication path shown in FIG. 1.

127. In sum, Ellis therefore purports to solve the problem of "sports-specific viewing and tuning enhancements" by implementing a sports watcher application for "more efficient sports tuning, especially in dual-tuner or multi-tuner systems." (EX-1016, ¶¶2; 75.) The sports watcher application provides an interactive scoreboard that "may be adjusted to include more or less information about each game" for which the user equipment (television) is presently tuned in. (EX-1016, ¶71.) The sports watcher application provides the user with information about numerous currently tuned in games, while providing said user the opportunity to change the channel to one of the tuned in games, which may be prompted on the basis of the provided information. (EX-1016, ¶89; FIG. 5.) Notwithstanding the television channel switching, the interactive sports watcher application may operate to ensure that "the tuner tuned to the primary sporting event may persist on the primary sporting event while another available tuner tunes to the requested media content." (EX-1016, ¶75.) According to Ellis, this allows the "channel changing or flipping [to] be generally performed on a secondary tuner." (EX-1016, ¶75.)

B. Overview Of Spivey

128. Spivey discloses an apparatus directed to distributing live event data to users in "real-time" utilizing a message queue server (MQS). (EX-1017, Abstract.) Unlike the '697 patent, Spivey does not concern video delivery. Nor does Spivey concern the technical problem addressed by the '697 patent; namely, the

synchronized delivery of real-time, contextually matched and actionable event information across multiple viewer client devices with an associated video stream.

129. In discussing certain conventional systems, Spivey describes certain "obstacles" to providing real-time event data to mobile devices. (EX-1017, ¶2.) Spivey's identification of said obstacles apparently centers on the latency drawbacks associated with storing information in a database. (EX-1017, ¶27.) According to Spivey, conventional live data content providers that use a database in connection with providing said live data often experience substantial delays, which may be "on the order of seconds (e.g., 30 to 45 seconds) or minutes." (EX-1017, ¶28.) For example, such conventional live data content providers may operate by first writing received data to a database, and then by subsequently waiting for clients to ask for the written data before ultimately sending the data to the client, thereby causing such delays. (EX-1017, ¶28.)

130. Within the context of this identified problem with conventional systems, Spivey seeks to provide real-time event data to mobile devices without a database. (*E.g.*, EX-1017, ¶51; *see generally*, EX-1017.) In this regard, Spivey discloses a system that—rather than storing live data—pushes live data to a client device "substantially simultaneously" with the receipt of said live data from a data source. (*E.g.*, EX-1017, Abstract, ¶¶51, 131.) To facilitate the immediate "push" of live data (sans database), Spivey discloses a message queue server that operates to

receive, sort, and push out data payloads "immediately," without interim storage of the data payload in a database. (EX-1017, ¶¶36, 51, 46, 128, 131.) I discuss this message queue server in further detail below.

131. But first, as a threshold matter, the message queue servers disclosed in Spivey are not described as a data source, nor do the message queue servers include stored live data. (*See generally* EX-1017.) Indeed, the very purpose of Spivey's described message queue server is to facilitate the transfer of live data to a client device without the need to store said data in a database. (*E.g.*, EX-1017, ¶51; *see generally*, EX-1017.) Rather than store data or otherwise constitute a data source itself, Spivey clarifies that the "[m]essage queue server 120 may ***receive live data from one or more live data sources 140***." (EX-1017, ¶36 (emphasis added).) The message queue server receives live data from the one or more live data sources so that the message queue server can "***process live data*** to provide live data quickly to the client devices" via the live data access server, as shown in the example of Spivey FIG. 1. (EX-1017, ¶36 (emphasis added); FIG. 1.)

132. Spivey's disclosure therefore unambiguously establishes that the message queue server receives, processes, and further relays data as an intermediary between live data sources and client devices; it does not constitute a data source itself, which is a separately described component in Spivey. (*See generally* EX-1017.) For example, Spivey describes the live data source as including "data feeds,

generated by computing devices associated with a live event" and/or "data gathered by human observers of a live event and entered into an interface." (EX-1017, ¶37.) In either case, the message queue server monitors and receives said data from the live data sources for subsequent processing and immediate "push" delivery. (EX-1017, ¶37; FIG. 1.)

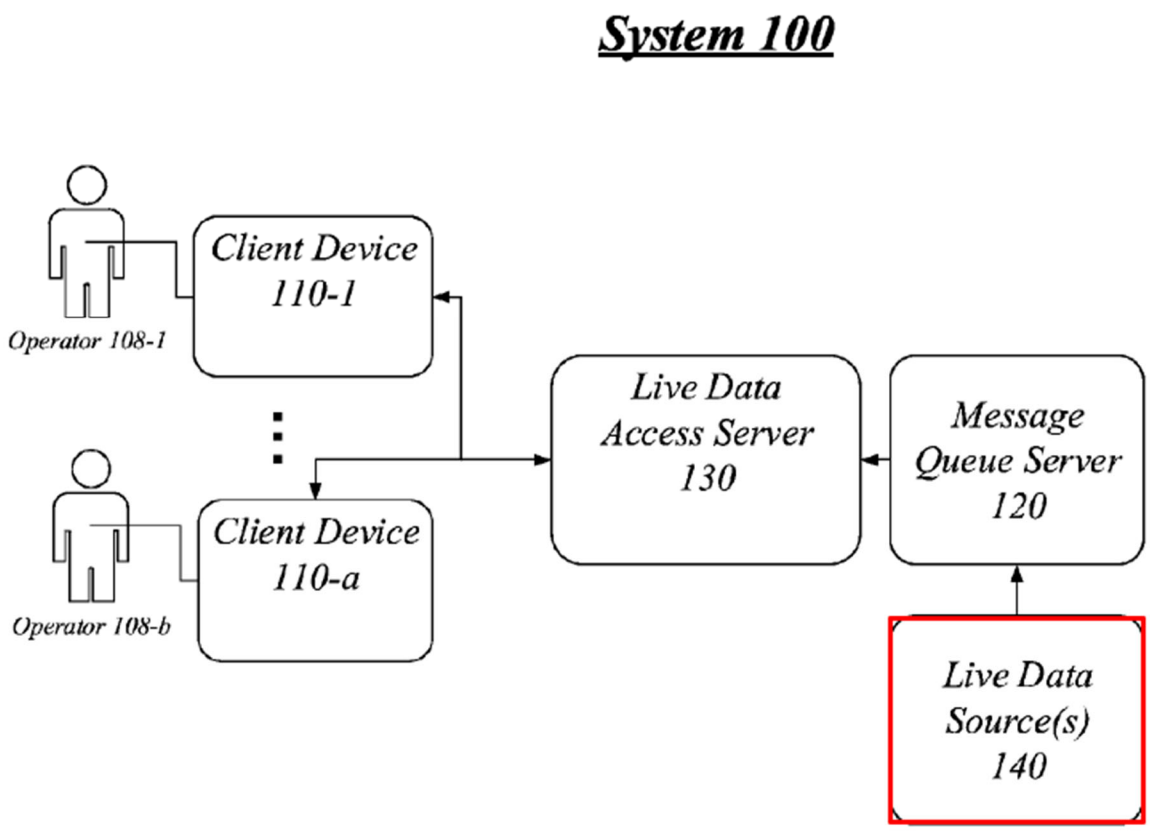


FIG. 1

(EX-1017, FIG. 1 (annotated).)

133. Spivey discloses the message queue server receiving "a unit of live data," including "a live data feed or a live data entry," and subsequently processing said data to facilitate delivery to downstream client devices. (EX-1017, ¶48.) The unit of data may include "information about at least one of a live event . . . an occurrence within a live event (e.g., a play, an error, an injury, a substitution, and so forth), and/or a participant at the live event (e.g., a player, a coach, an official, and so forth)." (EX-1017, ¶48.)

134. Spivey describes receiving the unit of live data as part of a "data payload." (EX-1017, ¶48.) The data payload includes two components: (i) "a topic label," and (ii) "a data payload." (EX-1017, ¶48.) The data payload includes said unit of live data. (EX-1017, ¶48.) The topic label may correspond to a topic queue of the message queue server and may therefore provide an indication of the topic queue of the message queue server with which the data payload should be associated. (EX-1017, ¶¶48, 49.)

135. Spivey describes the message queue server functioning to first sort the data payload into a designated queue based on the topic label, and then to push the data payload from the data queue to a live access server 130. (EX-1017, ¶¶47, 67.) For example, the message queue server may include a "topic decoder" that "remove[s] the topic label from the topic entry, use[s] the topic label to identify a topic queue." (EX-1017, ¶49.) Further, the topic decoder may be responsible for

"plac[ing] the data payload of the topic entry into the identified topic queue." (EX-1017, ¶49.) Once the data payload is placed in the appropriate topic queue, the message queue server operates, using, for example a MQS communication component, to "***push*** the data payload from the topic queue ***to a live data access server that has subscribed to the topic of the topic queue***." (EX-1017, ¶49; FIG. 1.) In this regard, the message queue server may cause the data payload to arrive at the downstream live data access server without saving the data payload in an intervening database. (*See, e.g.*, EX-1017, ¶68; *generally* EX-1017.)

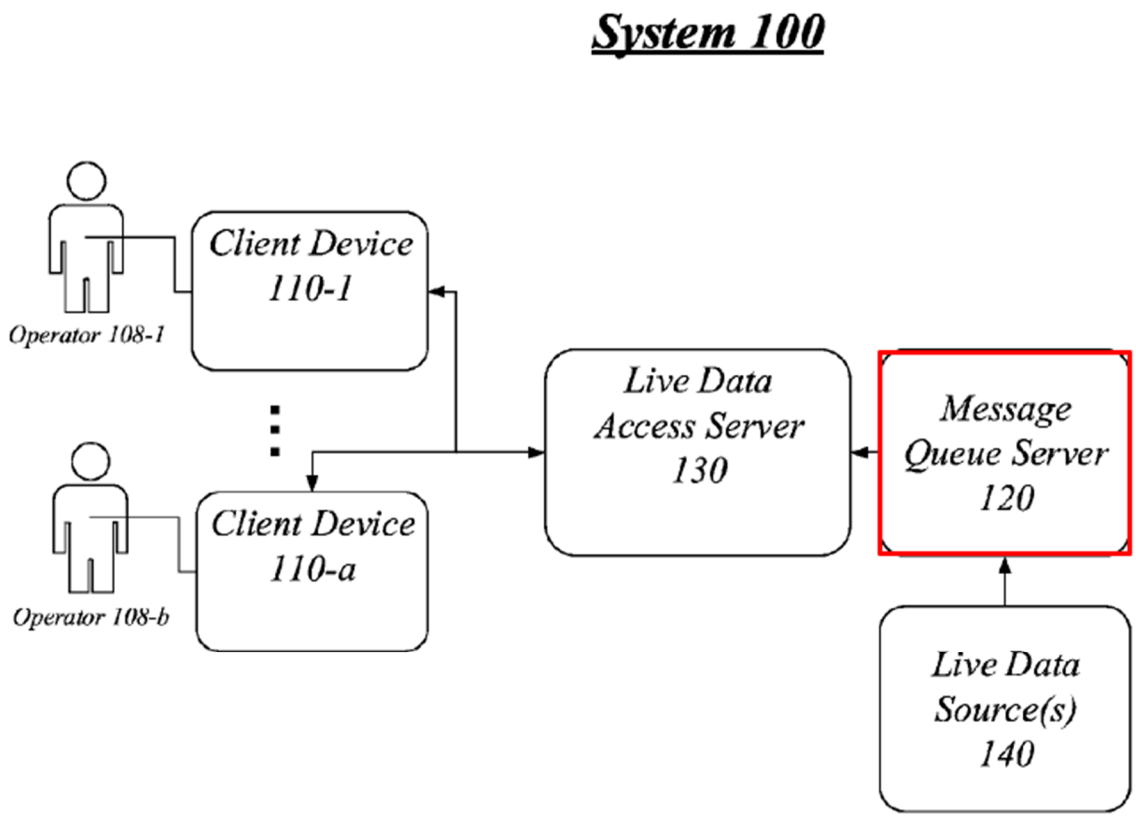


FIG. 1

(EX-1017, FIG. 1 (annotated).)

136. Spivey describes using the message queue server to send data payloads to certain "subscribers" of the topic queues, such as the live data access server. (EX-1017, ¶¶49, 128, 131.) For example, "[o]nce the data payload is inserted into a topic queue, message queue server 1420 may push the data payload to subscribers of the topic." (EX-1017, ¶ 128.) Spivey describes the subscribers of the topic queue as including the "aggregator component," as well as the "live data access server." (EX-1017, ¶ 128, 131.)

137. Absent from Spivey is any disclosed embodiment by which the message queue server communicates directly with client devices. (*See generally* EX-1017.) Indeed, in the principal Spivey embodiment of FIG. 1, Spivey teaches that the live data access server provides a public interface between client devices and message queue server 120. (EX-1017, ¶38.) Spivey does not discuss any embodiment that establishes a socketed connection between client devices and the message queue server, sans the live data access server as an intermediary. As set forth in Spivey:

> For example, live data access server 130 may allow operators 108 of client devices 110 to select topics related to a live event to subscribe to, and may retrieve the topic information and push the information to the client devices

> 110…. A client device, e.g. client device 110-1, may be usable by an operator, e.g. operator 108-1, to subscribe to and receive live data ***via live data access server 130***.

(EX-1017, ¶¶38-39 (emphasis added).)

138. Acting as such public interface, Spivey describes the live data access server as facilitating the access of client devices to the topic queues of the message queue server. (EX-1017, ¶38.) Spivey explains that the live data access server "allow[s] operators 108 of client devices 110 to select topics related to a live event to subscribe to, and [to] retrieve the topic information and push the information to the client devices 110." (EX-1017, ¶ 38.) In this manner, a client device may "subscribe" to the topic queue of the message queue server via the live data access server, but only the live data access server itself is considered the "subscriber" to the message queue server, according to Spivey. (EX-1017, ¶¶ 36, 38.)

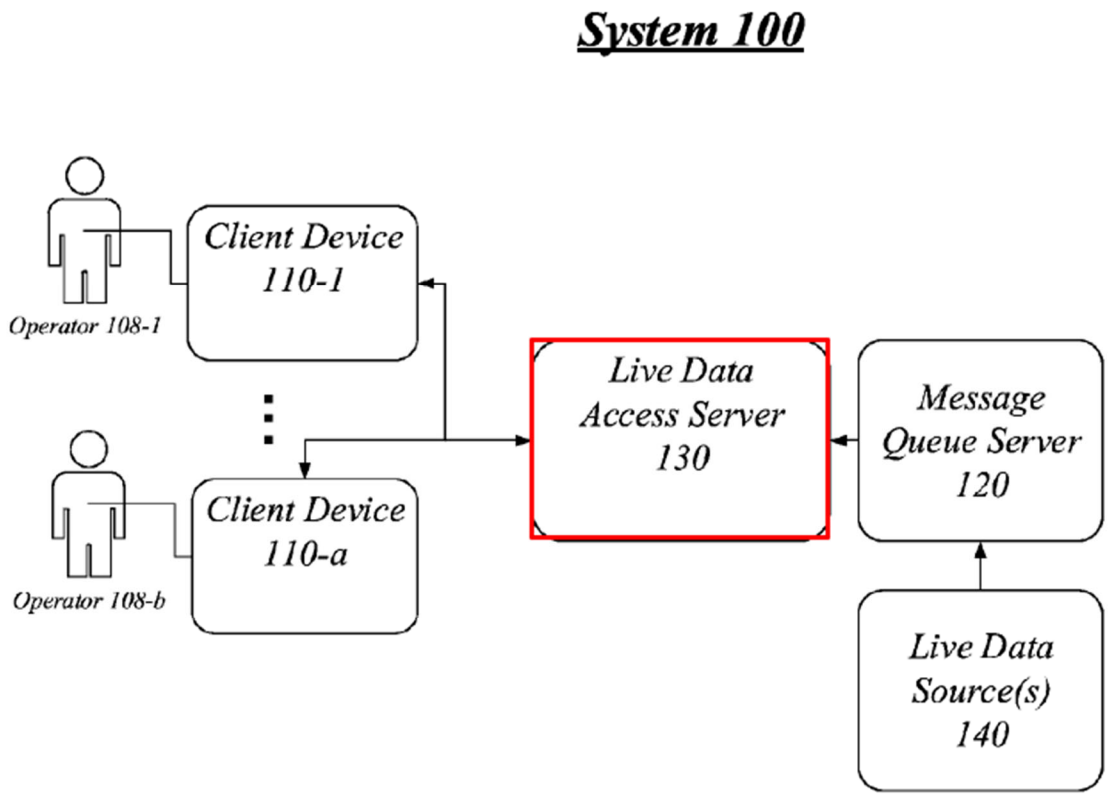


FIG. 1

(EX-1017, FIG. 1 (annotated).)

139. Spivey describes the live data access server as providing a "websocket connection with a client device." (EX-1017, ¶61.) The websocket connection between the live data access server and the client device may be "a two-way connection between two entities, where each of the two entities can send and receive data simultaneously, analogously to a voice telephone connection where both parties can talk and listen at the same time." (EX-1017, ¶61.) The only mention in Spivey of a websocket connection is in relation to the connection between the client devices

and the live data access server; Spivey does not describe the message queue server as establishing, or being capable of establishing, a websocket connection with any other component of Spivey system. (*See generally* EX-1017.) As I discuss further below, Spivey describes the message queue server as implementing a one-way message flow. (EX-1017, ¶¶65-70; FIG. 6.)

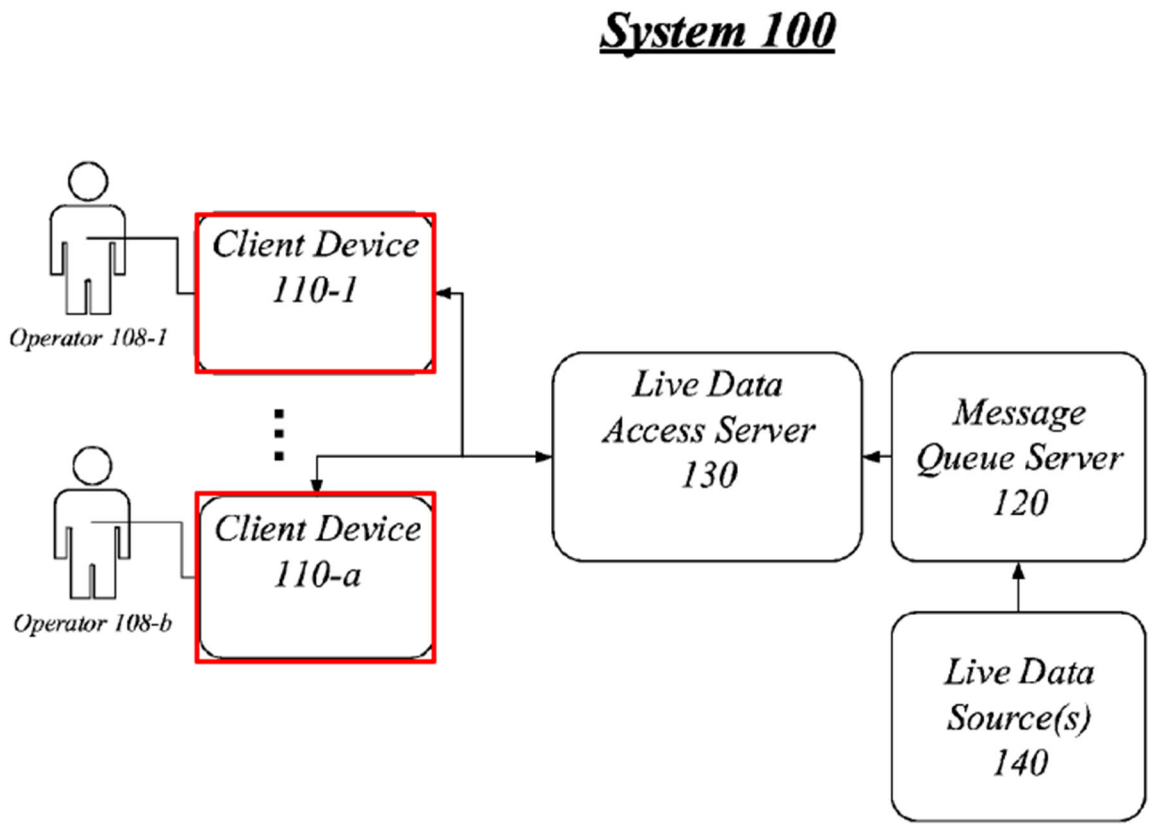


FIG. 1

(EX-1017, FIG. 1 (annotated).)

140. Within the context and confines of the functions described in relation to the system 100 of FIG. 1, Spivey describes that the message queue server 120 may be implemented with "two separate devices: [i] a message queue server device 410 and [ii] a live data server device 450." (EX-1017, ¶53; FIG. 4.) Spivey recognizes that "[i]mplementing a message queue server as two separate devices may be useful to protect proprietary data that may be stored on live data server device 450 and/or to limit potentially harmful access to the live data server device." (EX-1017, ¶57.) Notwithstanding, the message queue server device disclosed in the alternative FIG. 4 embodiment operates "substantially similar to message queue server (MQS) device" described in relation to the primary Spivey embodiment of FIG. 1. (EX-1017, ¶54.)

141. Accordingly, rather than deviate from the Spivey primary embodiment message queue server teachings, Spivey describes the live data server device (i.e., ***device one*** of the two-separate-device message queue server) as including a topic decoder that "operate[s] analogously to the topic decoder" described in relation to the primary Spivey embodiment of FIG. 1. (EX-1017, ¶56.) In this regard, the live data server device (device one) operates to receive a topic entry, strip a topic header of said topic entry, and sort the associated data payload into a topic queue of the live data server device, analogous to the topic decoder of the primary embodiment. (EX-1017, ¶¶47-49; 56.)

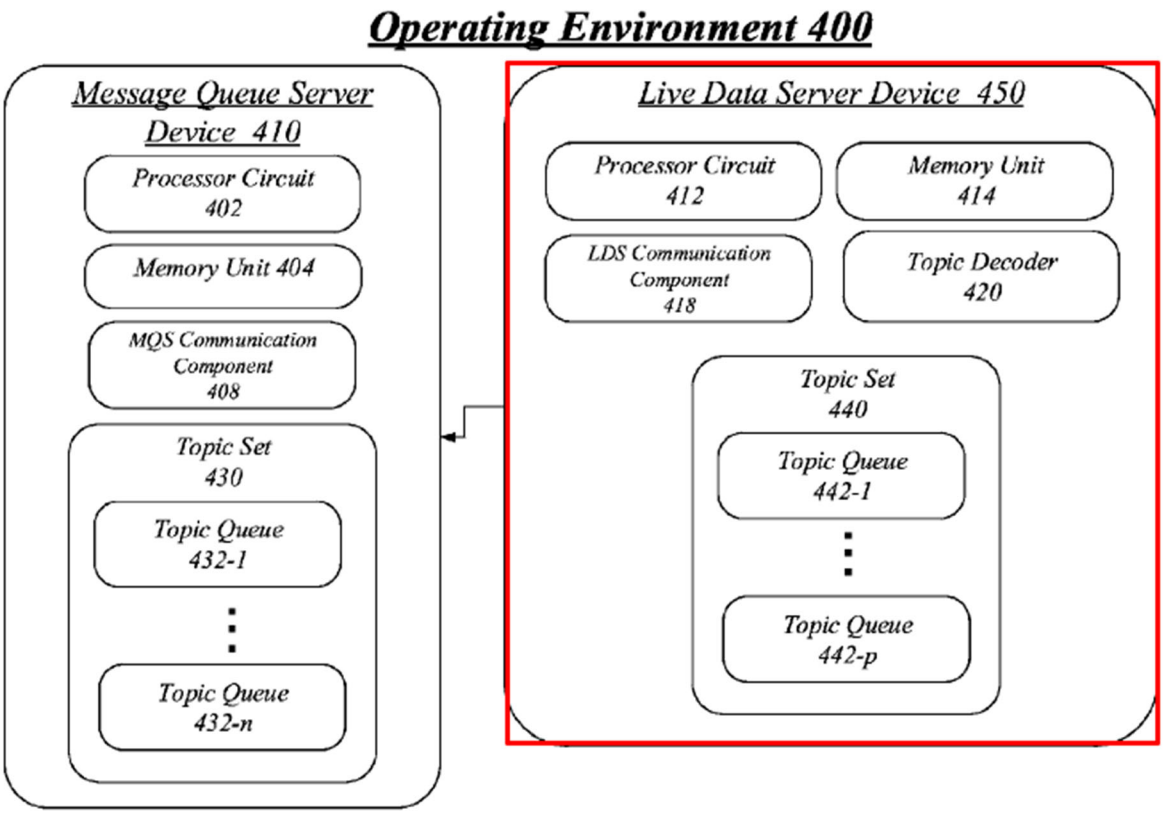


FIG. 4

(EX-1017, FIG. 4 (annotated).)

142. Rather than send the data payload to a subscriber device, the live data server device sends the data payload to the message queue server device (i.e., ***device two*** of the two-separate-device message queue server). (EX-1017, ¶¶56-58.) Spivey explains that the message queue server device (device two) operates as a "message broker software" that "facilitates topic queues that receive published messages" in support of the "push" of said messages downstream. (EX-1017, ¶58.) For example, the message queue server device includes an MQS Communication Component 408, which Spivey describes as being "analogous . . . to . . . ***MQS communication***

component 308." (EX-1017, ¶54 (emphasis added).) Accordingly, a POSITA would understand the MQS component 408 of the Spivey FIG. 4 alternative embodiment to have the same functionality as the MQS communication component 308; namely, to "***push*** the data payload from the topic queue ***to a live data access server that has subscribed*** to the topic of the topic queue." (EX-1017, ¶49 (emphasis added).)

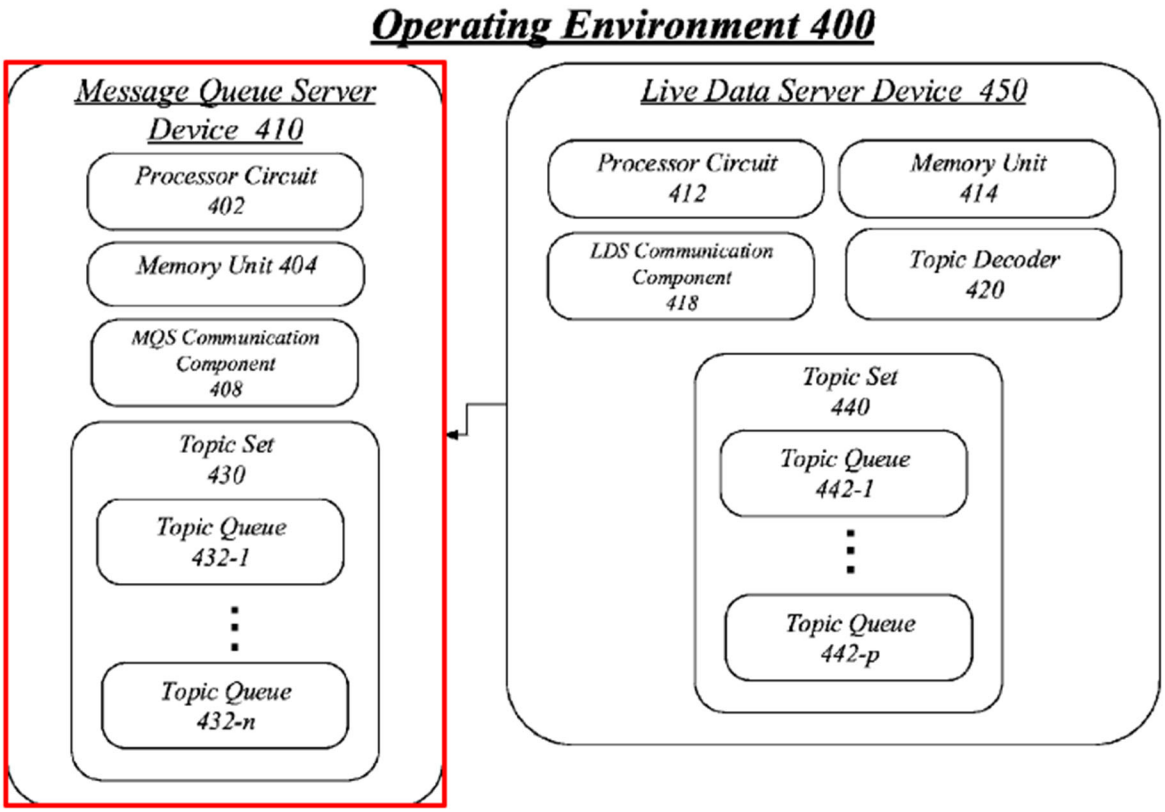


FIG. 4

(EX-1017, FIG. 4 (annotated).)

143. Spivey consistently describes a one-way communication flow from live data sources, on the one hand, to client devices, on the other. (*E.g.*, EX-1017, ¶65; FIG. 6.) The message queue server and live data access server facilitate said one-

way communication flow. (*E.g.*, EX-1017, ¶65; FIG. 6.) For example, Spivey depicts an example communication or "message flow" in FIG. 6 whereby "live data is received from live data source(s) 140 at message queue server 120 in message 602." (EX-1017, ¶66; FIG. 6.) The initial message (i.e., message 602) is in the "form of a topic entry, which may include, at least, a topic label and a data payload." (EX-1017, ¶66; FIG. 6.) Upon receiving said message 602, the message queue server strips and sorts the topic entry. (EX-1017, ¶67; FIG. 6.) Spivey goes on to describe the operation of message queue server sending a message 606 to the live data access server (e.g., "message queue server 120 may push the data payload to live data access server 130"), and the operation of live data access server sending a message 608 to one or more client devices (e.g., "live data access server 130 may push the data payload to client device 110-a"). (EX-1017, ¶68, 69; FIG. 6.) Notably, each message (e.g., messages 602, 604, 606, 608) represents a one-way flow of a "unit of live data" between the live data source and, ultimately, the downstream client devices. (EX-1017, ¶¶48, 67; FIG. 6.)

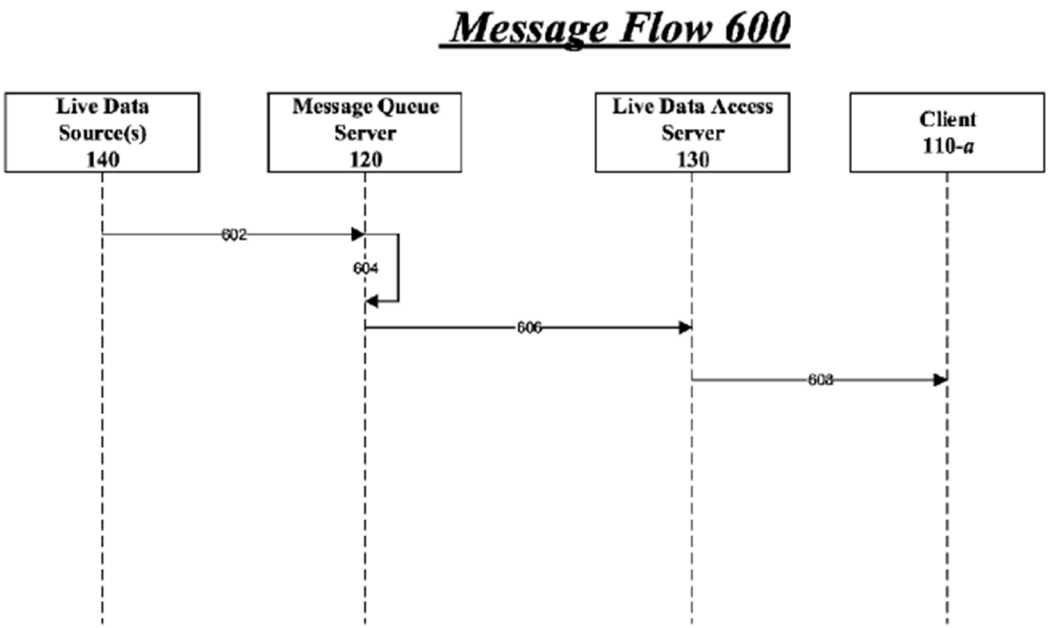


FIG. 6

144. In sum, Spivey purports to solve the alleged problem of conventional live event data providers latency by offering a database less system, in which a message queue server and associated live data access server facilitate a one-way flow of data payloads between a live data source and one or more client devices. (*See generally* EX-1017.)

C. Overview Of Herzog

145. Herzog discloses methods and software applications for separately transmitting video and data streams from a live event and then assembling those streams together by matching the streams timestamps. (EX-1015, Abstract, ¶¶18,

53-60.) Unlike the ’697 patent, Herzog does not disclose a socket server of any type or manner. Nor does Herzog concern the technical problem addressed by the ’697 patent; namely, the synchronized delivery of real-time, contextually matched and actionable event information across multiple viewer client devices with an associated video stream.

146. In discussing the “expand and increased popularity” of wireless mobile communications, including data speed, Herzog identifies the desirability of “provid[ing] a mobile device, such as a tablet, with a capability of receiving multiple sources of live information that are related to each other, but are published and processed at different rates.” (EX-1015, ¶3.) For example, Herzog recognizes that some sources of live information may include both “video” and “meta-data, [which] provides additional information related to the HLS video.” (EX-1015, ¶3.) The illustrative use case described in Herzog for such live information, which I discuss in further detail below, is an Indy car race, in which a “source” may include both video of a racecar driver and associated meta-data pertaining to the driver’s racecar. (EX-1015, Abstract, ¶54.) Despite the foregoing video and meta-data being related to the same source of live information, each may be “published and processed at a different rate.” (EX-1015, Abstract, ¶3.) According to Herzog, at the time of the Herzog disclosure, the assembling of such video and meta-data at a mobile device

"for simultaneous display, in real-time has not as yet been implemented." (EX-1015, Abstract, ¶3.)

147. Within the context of this identified problem of matching video and associated meta-data streams at a mobile device, Herzog seeks to match the audio / video stream and the meta-data stream with one another using a GPS-created time stamp associated with each stream. (EX-1015, ¶¶18, 53, 55, 56.) According to Herzog, "the meta-data are both time stamped with consistent (e.g., GPS) timing protocols" such that "each respective data is time-tagged to the same timing signature, even though each respective data transmits from different physical locations." (EX-1015, ¶18.) As described in Herzog, "[t]wo time stamps now exist in the system"—one associated with the video, and one associated with the meta-data. (EX-1015, ¶56.) At the mobile device, the audio/video stream "is deemed to be the master time for synchronization" with the now time stamped meta-data "slaved to the master time generate by the VITC [video/audio] time code." (EX-1015, ¶¶54, 58.)

148. Herzog describes the foregoing time stamp matching techniques in connection with a single event- an Indy race. (EX-1015, Abstract; ¶18.) While Herzog clarifies the Indy race is "example" event, Herzog fails to provide any other examples of possible events. (*See generally* EX-1015.) Notwithstanding, Herzog clearly contemplates all of the audio / video streams and meta-data streams deriving

from the "same event." (EX-1015, ¶¶27, 36, 80, 81.) That is, Herzog describes that all of the audio / video streams and all of the meta-data streams originate from, or otherwise transmit from, "the same venue." (EX-1015, ¶¶16, 54)

149. With respect to the only specific example venue or event described in Herzog—the Indy race—Herzog describes and shows a collection of sources (e.g., sources 48a – 48d) that represent sources of live information. (EX-1015, Abstract, ¶¶46, 47, 54; FIG. 1.) Herzog describes that "our video/audio streams provided from four different imaging sources, namely sources 48a-48d.." (EX-1015, ¶46.) The video / audio from each source may be captured with "a camera positioned inside a race car," as an example. (EX-1015, ¶46.) In this regard, a POSITA would understand that the Herzog disclosure contemplates each of the sources as a racecar, for example, so that a different video/audio stream may be captured from each source/racecar. (*See generally* EX-1015.)

150. Herzog further recognizes that telemetry data (meta-data) may be captured from each of the multiple sources. (EX-1015, ¶47.) That is, the same "source" of live information may be a source of both captured video/audio and meta-data. (EX-1015, ¶¶3, 47.) Herzog continues the Indy race example by noting that "telemetry is widely used in motor car racing, allowing race engineers to interpret data collected during a race and use it to properly tune a race car for optimal

performance." (EX-1015, ¶47.) Accordingly, Herzog recognizes that such sources "may include audio/video data, meta-data, or ***both types of data***." (EX-1015, ¶47.)

151. Despite the source including, or being associated with capture of, both types of data (i.e., both audio/video data and meta-data), Herzog contemplates transmitting and processing each data type differently. I.e., audio / video data is transmitted utilizing a specific data format and interface, while meta-data is transmitted as a different data format. (*See generally* EX-1015.) For example, the audio / video data, upon capture, "are transmitted to a satellite truck 51 that is located adjacent the event (Indy Race, for example)." (EX-1015, ¶46.) The satellite truck is configured to transmit the audio/video data "in the form of an asynchronous serial interface (ASI) format, which streams data carrying an MPEG transport stream (MPEG-TS)," to, for example, a satellite (EX-1015, ¶46; FIG. 4.) In this preferred embodiment, Herzog only specifies one data format through one interface (ASI to transmitting MPEG-TS data). Herzog did not specify coping the audio / video data and transmitting the audio / video data via several different data formats through varied media sources.

152. Herzog describes transmitting the meta-data from the venue separately from the audio/video data. (*See generally* EX-1015.) For example, Herzog describes "an Enhanced Result Protocol (ERP) generator server 49 which is configured to receive telemetry data (meta-data) from the multiple sources 48 a-48 d." (EX-1015,

¶47.) The Enhanced Result Protocol generator server formats the captured meta-data "into a Hypertext Transfer Protocol Secure (HTTPS) communications protocol for secure communication over the network." (EX-1015, ¶47.)

153. As represented in Herzog's system diagram of FIG. 4, the audio/video data is transmitted separately from the venue separately from the meta-data. (EX-1015, FIG. 4.) Indeed, Herzog clarifies that "the physical geographical location of the video/audio transmission point and the geographical location of the meta-data server may be ***up to a mile apart within the same venue***." (EX-1015, ¶54 (emphasis added).) Herzog also clarifies that the audio/video data and the meta-data may be transmitted from the venue under different protocols (e.g., ASI vs. HTTPS), and may each exhibit other distinct characteristics, such as distinct latency characteristics. (EX-1015, ¶¶46, 47, 64.)

154. Despite the different physical transmission points, and despite the different characteristics of the transmitted data, Herzog proposes a technique to later match said transmitted audio/video data and meta-data using GPS time stamps. (EX-1015, ¶18.) For example, "the video and audio data and the meta-data are both time stamped with consistent (e.g., GPS) timing protocols." (EX-1015, ¶18.) Accordingly, "each respective data is time-tagged to the same timing signature, even though each respective data transmits from different physical locations." (EX-1015, ¶18.) Because each data type is time stamped at the "transmission point," each data

type can be later matched with one another according to the time stamp so that the time stamp provides a common point of reference as between the different audio/video stream and the meta-data stream. (EX-1015, ¶¶53, 55.)

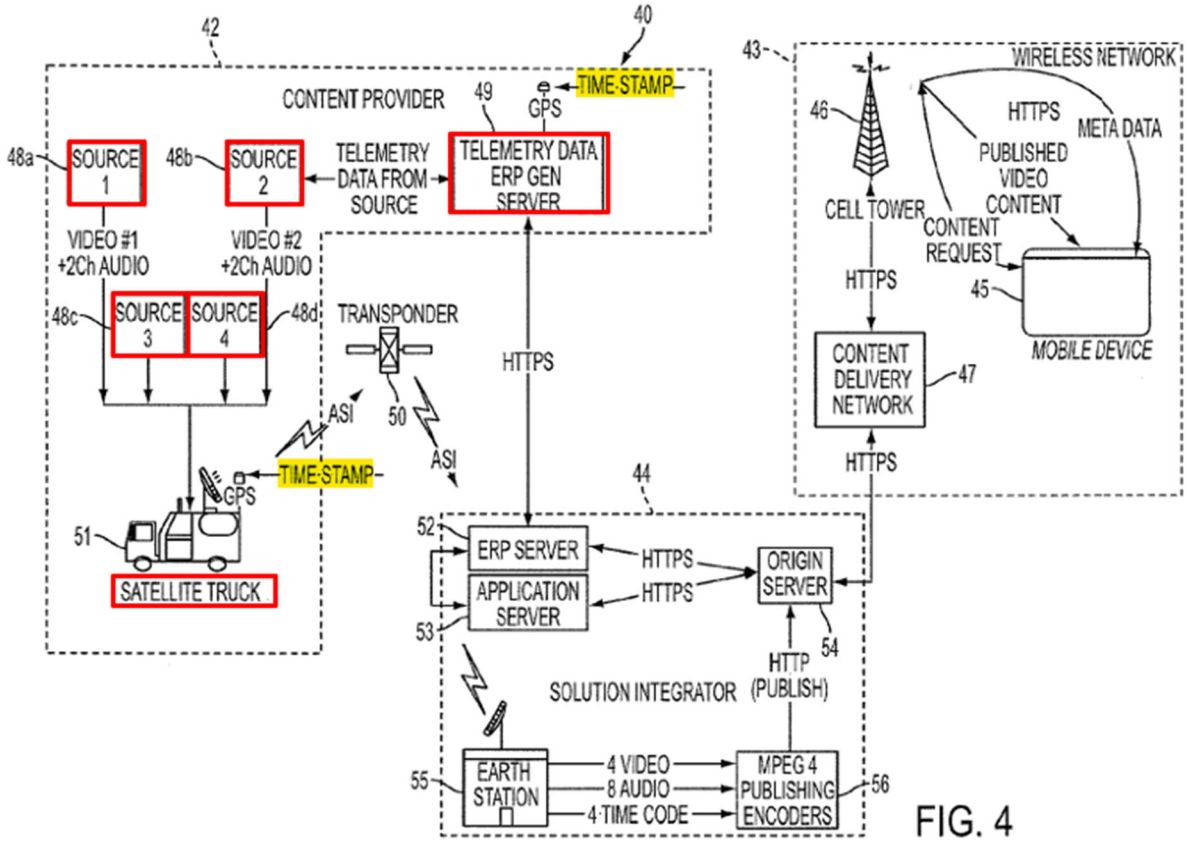


FIG. 4

155. Herzog discloses a "solutions integrator" to ingest each of the time stamped streams. (EX-1015, ¶55; FIG. 4.) With respect to the meta-data, Herzog's solution integrator "is configured to ingest the time stamped meta-data over IP into a process or origin type server system, namely, into the origin server 54 (for example)." (EX-1015, ¶55.) As shown with reference to Herzog FIG. 1, the meta-data proceeds to an origin server via an ERP Server, which "acts as an intermediary between the ERP generator server and the origin server" to "filter and eliminate

contents of meta-data that are race-sensitive, in that the contents may provide advantages to other race cars participating in the race." (EX-1015, ¶55; FIG. 1.) In this regard, the time stamped meta-data is filtered for sensitive information before reaching the origin server. (EX-1015, ¶55; FIG. 1.) From the origin server, the meta-data is sent to an external content delivery network in connection with the audio / video data. (EX-1015, ¶51; FIG. 1.)

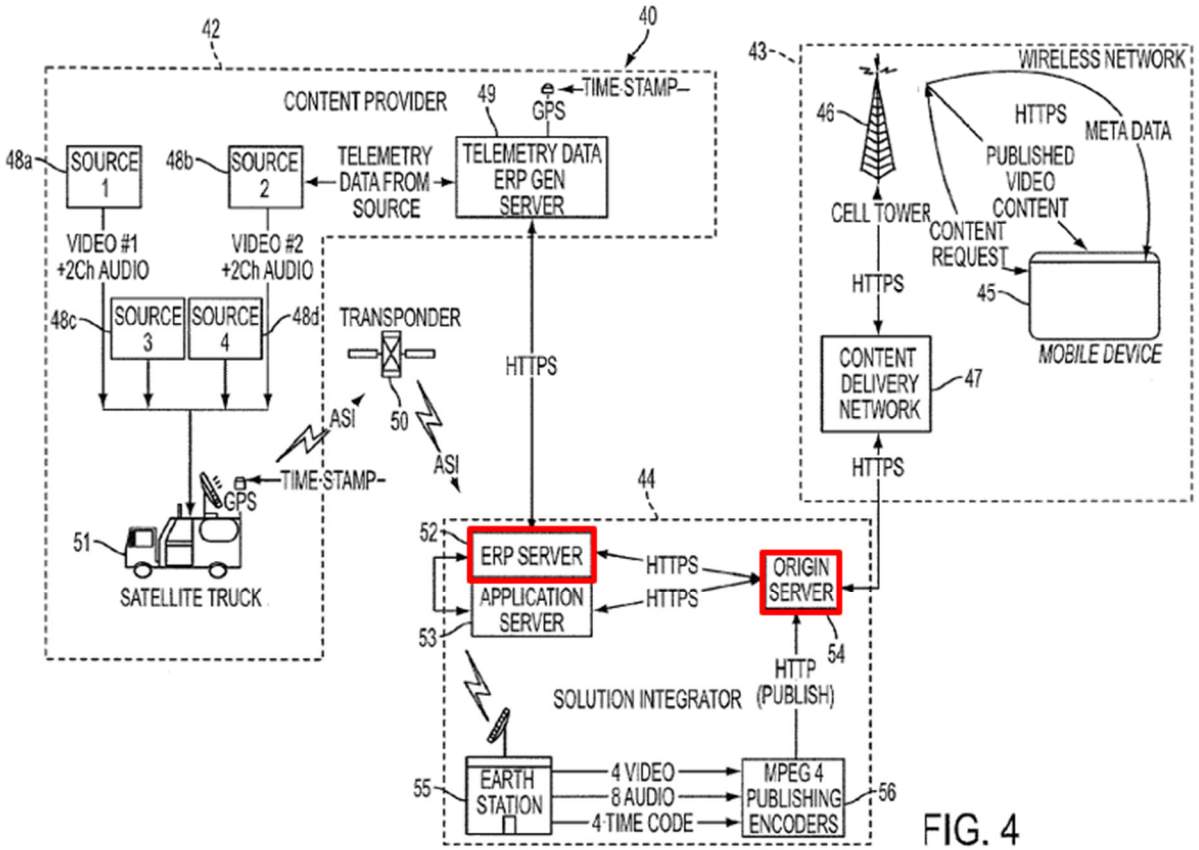


FIG. 4

156. With respect to the audio / video data, Herzog's solution integrator, similarly, "is configured to ingest the video, audio and VITC time code." (EX-1015, ¶55.) As shown with reference to Herzog FIG. 1, the audio / video data proceeds to

the origin server via an earth station and MPEG 4 Publishing encoders. (EX-1015, ¶¶48-49; FIG. 1.) The MPEG 4 Publishing Encoders encode the audio / video data for streaming to the origin server. (EX-1015, ¶49.) In this regard, the time stamped audio / video data is encoded in HTTP (published) form before reaching the origin server. (EX-1015, ¶51; FIG. 1.) From the origin server, the audio / video data is sent to an external content delivery network in connection with the meta-data. (EX-1015, ¶51; FIG. 1.)

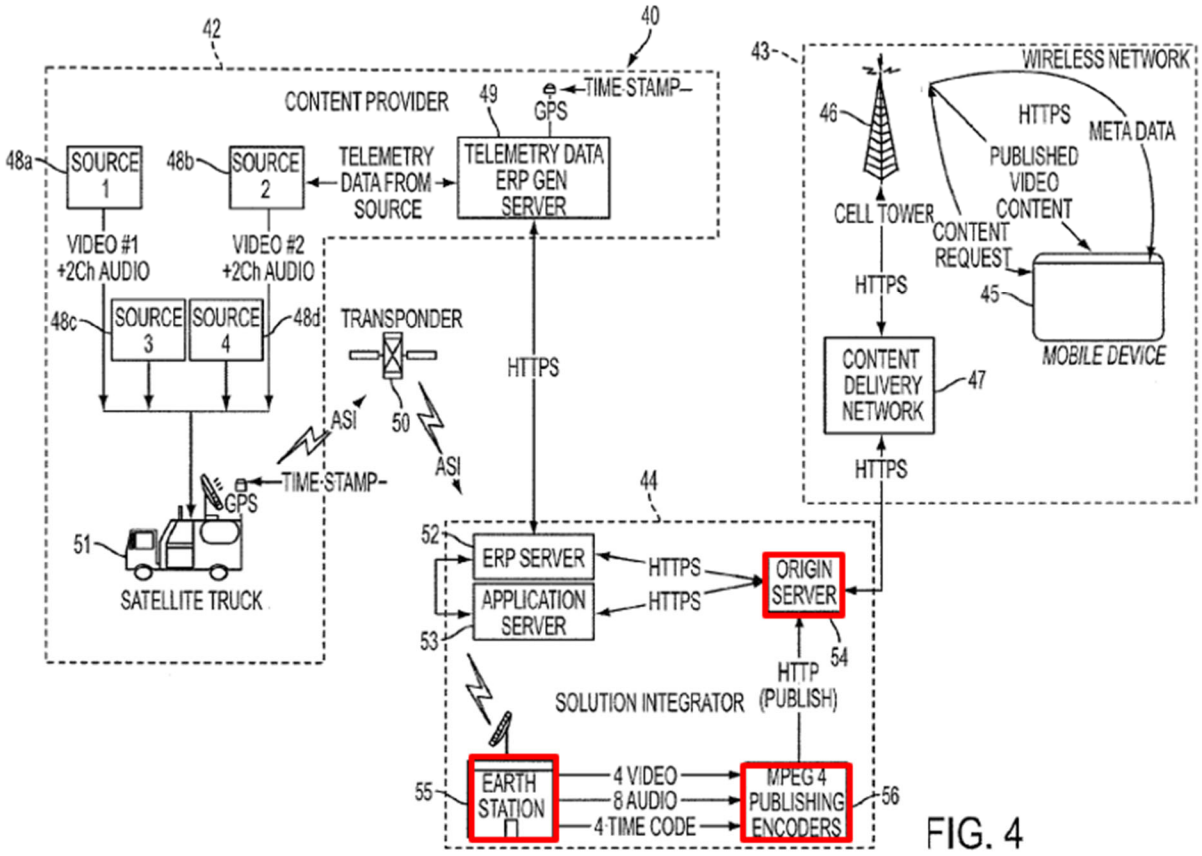


FIG. 4

157. Herzog describes transmitting the audio / video data and the meta-data from the solution integrator to a wireless network. (EX-1015, ¶¶43, 51; FIG. 1.) The wireless network includes both a content delivery network and a mobile device. (EX-1015, ¶43; FIG. 1.) The content delivery network receives the "video imagery, related audio and meta-data that are received from the content provider" and facilitates the delivery of said information to one or more mobile devices associated with said wireless network. (EX-1015, ¶49; FIG. 1.) For example, and as shown in Herzog FIG. 1, a mobile device may request (via a cell tower) published video content from the content delivery network. (EX-1015, ¶¶57-58; FIG. 1.) The published video content is delivered to the mobile device including a time stamped code. (EX-1015, ¶¶57-58.) The mobile device may then request meta-data associated with the video. (EX-1015, ¶58.) The meta-data also includes a time stamped code and can be paired with the time stamped coded video accordingly. (EX-1015, ¶58.)

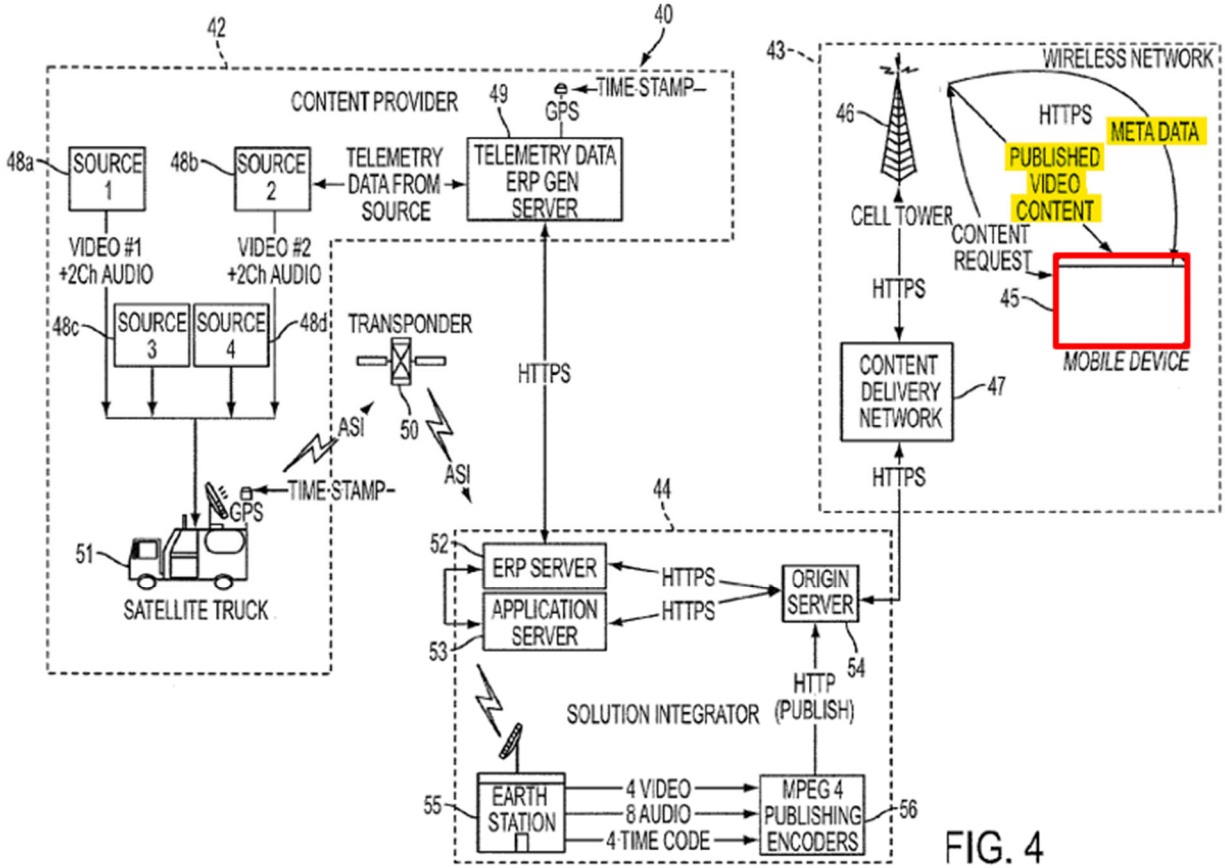


FIG. 4

158. Herzog contemplates sending the video / audio data and the meta-data to the mobile device using different protocols. (EX-1015, ¶¶59-61.) For example, the audio/video stream may be sent to the mobile device via an HLS protocol whereas the meta-data may be sent to the mobile device via an HTTP protocol. (EX-1015, ¶¶59-61.) For example, "[a] server (for example, a solutions integrator 67) receives the video/audio data as an RF downlink from the satellite truck 65 and separately receives the meta-data as an HTTPS formatted signal, e.g. via a terrestrial network providing IP packet data transport." (EX-1015, ¶61.) The video/audio data is propagated through the system to the mobile device as an HLS formatted signal,

whereas as the meta-data is propagated through the system to the mobile device as an HTTP formatted signal, which is a different type of signal as compared to the HLS formatted signal (EX-1015, ¶61.)

159. Only when the two separate streams arrive at the mobile device 70 are "[b]oth types of data [] assembled in the mobile device 45 via the client application software that resides within the mobile device, for appropriate presentation on the device's display." (EX-1015, ¶58.)

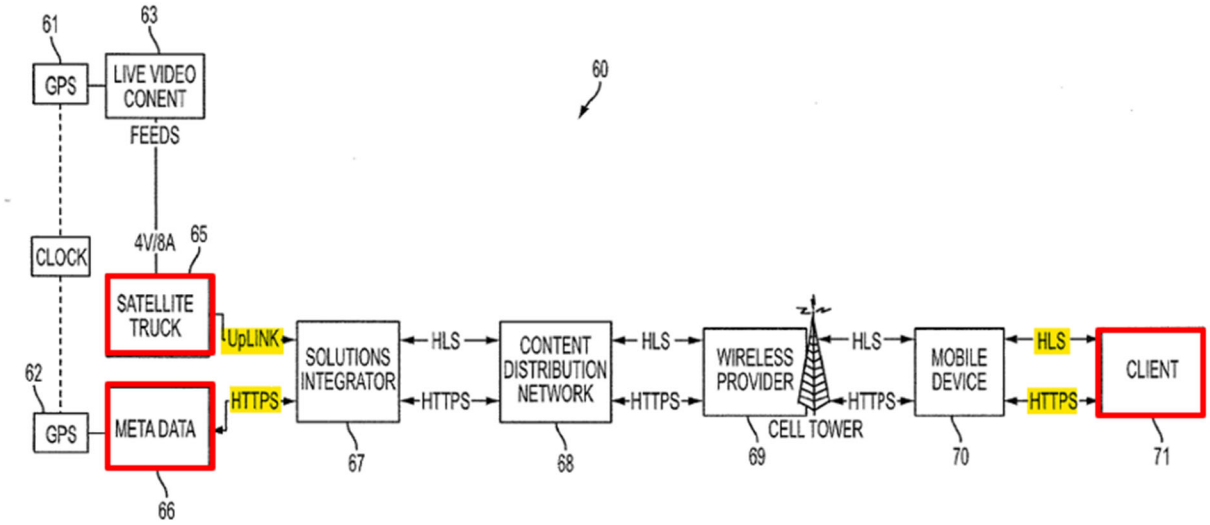


FIG. 5

160. For example, and in connection with its discussion of FIG. 6, Herzog recognizes that while the latency associated with a "live time stamped video" may be greater than 24 seconds, the latency associated with a "live time stamped meta data" may be less than 1 second. (EX-1015, ¶56.) Inherent in Herzog's disclosure is that audio/video data captured from a source will take longer to reach a mobile

device as compared to meta-data captured from the exact same source at the same time. (EX-1015, ¶¶56, 65.) Because of this difference in delivery time—i.e., latency—Herzog discloses a time stamp matching technique whereby "the time code associated with first type of data (for example, video data) becomes the master clock and the time code associated with the second type of data (for example, meta-data) becomes the slave clock." (EX-1015, ¶65.) Herzog continues, "the slave clock is slaved to the master clock" such that "the user now, advantageously, may view two types of data that normally are not viewable together because of their disparate latencies." (EX-1015, ¶65.) Accordingly, as between the audio / video data and the meta-data, the audio / video data sets the pace of transmission because it is slower. (EX-1015, ¶65.) And the faster-traveling meta-data is matched—"slaved to the master clock" of the audio / video data—to meet the audio/video data at the appropriate time stamped interval. (EX-1015, ¶65.) Therefore, a POSITA would understand that to the extent latency is a concern of Herzog, the fundamental issue is the latency of the slower arriving audio / video stream, and not the delivery time of the substantially faster-arriving meta-data.

161. To implement the foregoing time stamping matching technique, Herzog discloses a multistep process at FIG. 6 in which the depicted system implements a process for "listening" for the video stream, determining a time stamp or code of said video stream, and then retrieving the metadata with the corresponding time

stamp. (EX-1015, ¶¶70-77.) For example, on a request from the client application, "[t]he origin server 82 begins streaming HLS video of the live event with the VITC time stamp header to the client application 85." (EX-1015, ¶¶65-72; FIG. 6.)

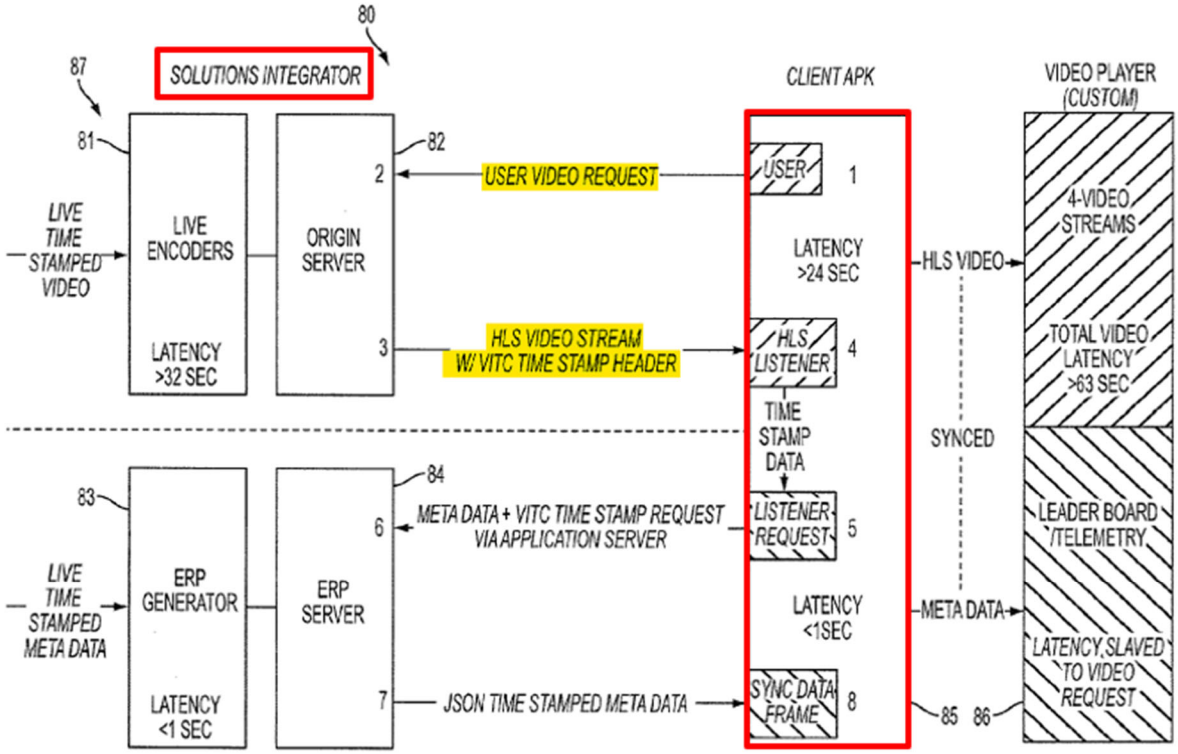


FIG. 6

162. Subsequently, Herzog describes, on a request from the client application, "transmit[ing] the meta-data with a corresponding .JSON time stamp of the live event" such that "[t]he client application receives the meta-data having the corresponding time stamp and displays the meta-data on the video player." (EX-1015, ¶¶65-77; FIG. 6.)

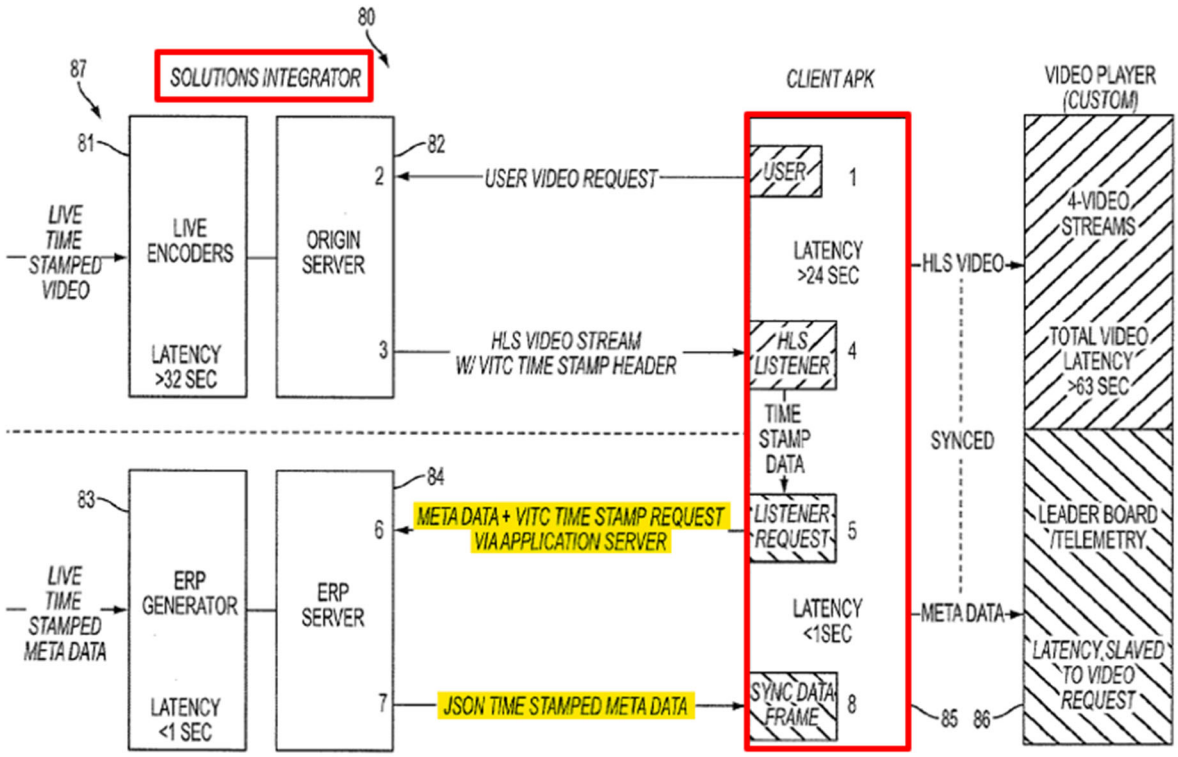


FIG. 6

163. Finally, the time stamped meta-data and audio / video data may be synced and presented to the user via a video player of the mobile device. (EX-1015, ¶¶77-79; FIG. 6.) For example, the client application may receive the meta-data with the corresponding time stamp and relate said meta-data to the video so that "the meta-data is now synchronized to the video channel." . (EX-1015, ¶77; FIG. 6.)

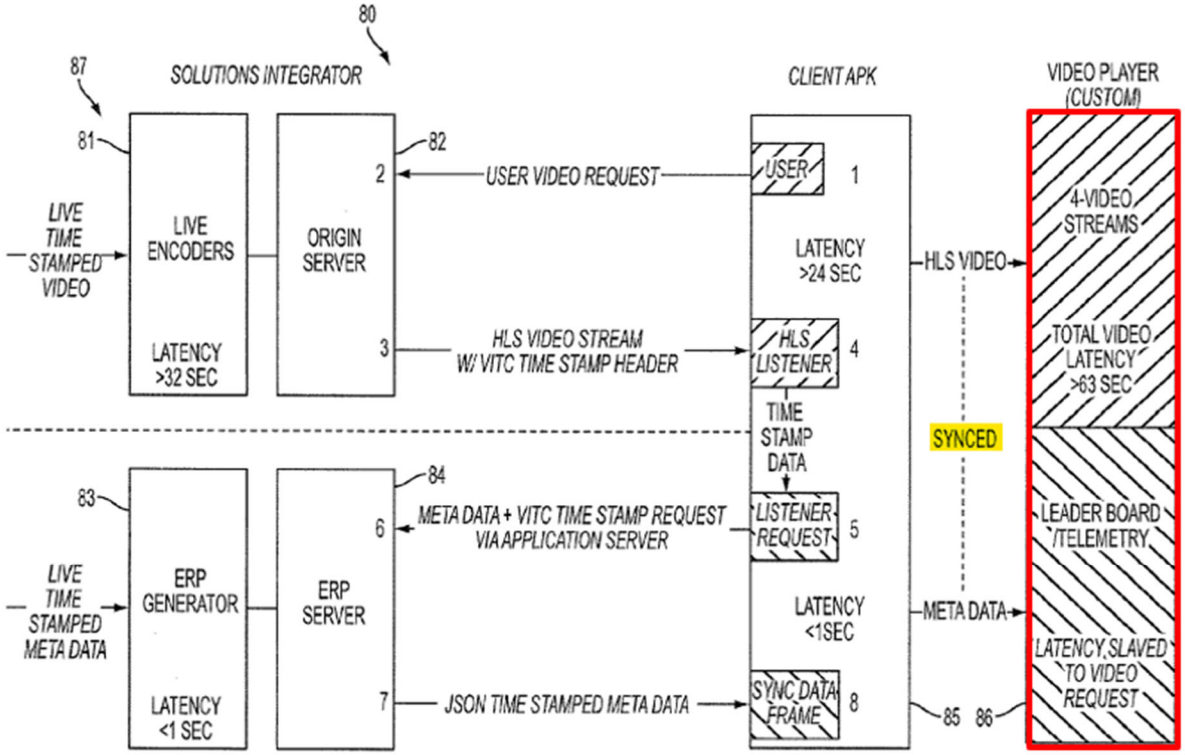


FIG. 6

164. Absent from Herzog is a discussion of much of the backend system architecture and/or other techniques for implementing the meta-data distribution described therein. (*see generally*, EX-1015.) Indeed, I agree with the conclusion of Petitioner's expert, Dr. Wicker, in that "Herzog does not specifically disclose some backend server architecture supporting the data distribution—the claimed socket server." (EX-1001, ¶126.)

165. In sum, Herzog purports to solve the alleged problem of matching disparate data streams by reference to a common touchstone of a GPS time stamp. (*see generally*, EX-1015.) The Herzog purported time stamp matching solution bears little relevance to the technical problem addressed by the '697 patent, including the

synchronizing delivery of contextually matched and actionable event information across multiple viewer client devices with an associated video stream.

VI. THE CITED PRIOR ART DOES NOT DISCLOSE OR RENDER OBVIOUS ALL LIMITATIONS OF INDEPENDENT CLAIMS 19, 23, AND 27

166. In this IPR2024-01309, Petitioner challenges three claim sets from the ’697 patent—claims 19-25, 27-30, of which, claims 19, 23, and 29 are the independent claims. I understand that Petitioner and its expert, Dr. Wicker, allege that two different combinations—Ellis in view of Spivey and Herzog in view of Spivey and Ellis—each render obvious independent claims 19, 23, and 27.

Ground 1	§103	Ellis in view of Spivey	Claims 19-25, 27-30
Ground 2	§103	Herzog in view of Spivey and Ellis	Claims 19-25, 27-30

167. I disagree with the arguments in the Petition and the opinions set forth by Dr. Wicker for at least the reasons that I discuss in further detail below.

A. Ground 1 Does Not Disclose Or Render Obvious All Limitations of Claims 19, 23, and 27

168. Ground 1’s challenge of independent claims 19, 23, and 27 relies on Petitioner’s assertion that a POSITA would have replaced a supplemental content source in Ellis with multiple servers disclosed by Spivey. (Petition, 19-24.)

169. First, a POSITA would not have been motivated to make such a combination when doing so would render the resulting system non-functional and

moot the very benefits that each reference sought to achieve. Second, even if the teachings of Ellis and Spivey are combined as Petitioner suggests, the resulting system would not disclose or render obvious transmitting information via a channel between an "*event socket*" and a viewer client device.

170. Accordingly, Ground 1 does not disclose or render obvious all limitations of claims 19, 23, and 27[4].

1. A POSITA would not have been motivated to combine the teachings of Ellis and Spivey

171. In discussing Dr. Wicker's opinions regarding the combination of Ellis and Spivey, I will first address the proposed combination itself and the problems that arise from such a combination of teachings. I will then address the lack of motivation a POSITA would have had to make such a combination.

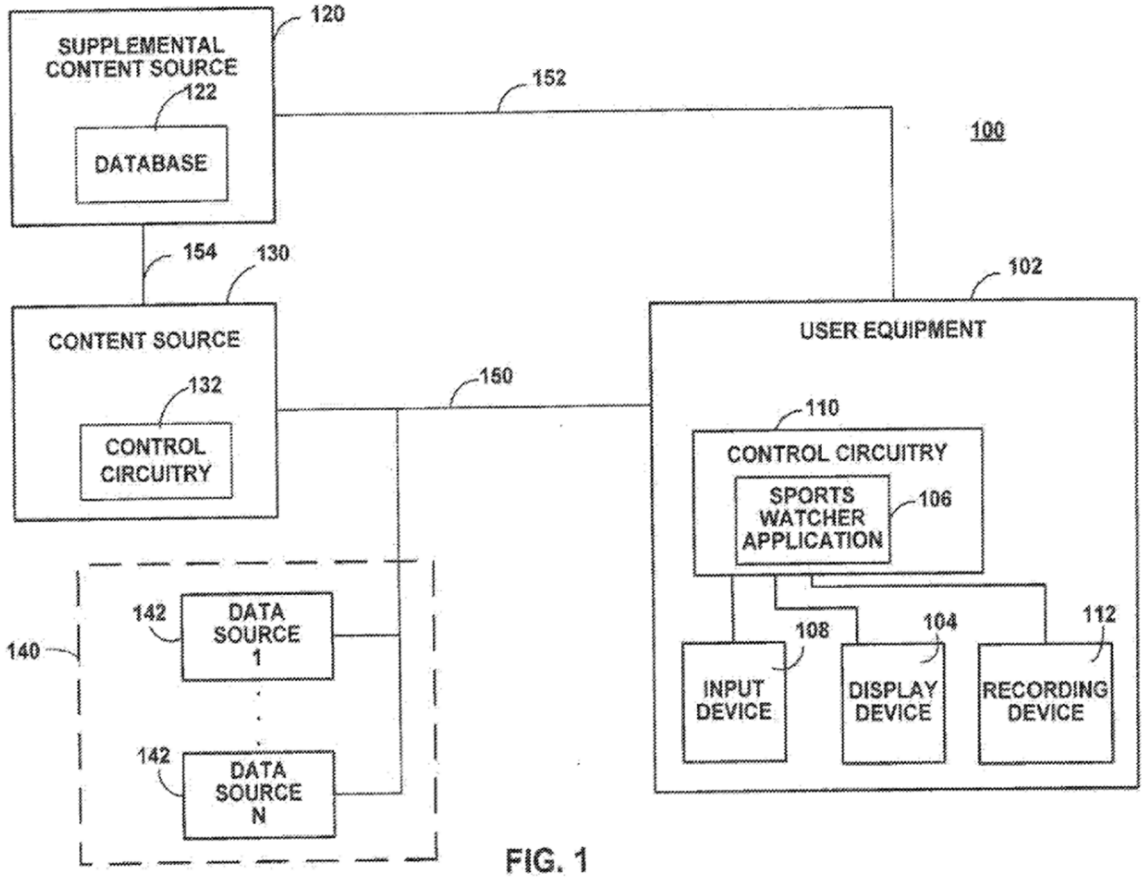
a. Petitioner's proposed combination radically modifies Spivey's teachings.

172. Petitioner and Dr. Wicker argue that a POSITA would have implemented a single component of Spivey's system—i.e., the message queue server, in place of Ellis's supplemental content source. (EX-1001, ¶¶111.) But Petitioner fails to articulate a way in which these cherry-picked disclosures could be

[4] I note that Dr. Wicker's analysis of independent claims 23 and 27 refers back to his analysis of claim 19. (EX-1001, pp. 393-395, 400-403.)

combined into an operational system. Even if one were to assume that such disclosures could be combined into an operational system, which is not taught by the prior art, the resulting system would lack the very benefits that Ellis and Spivey independently sought to achieve.

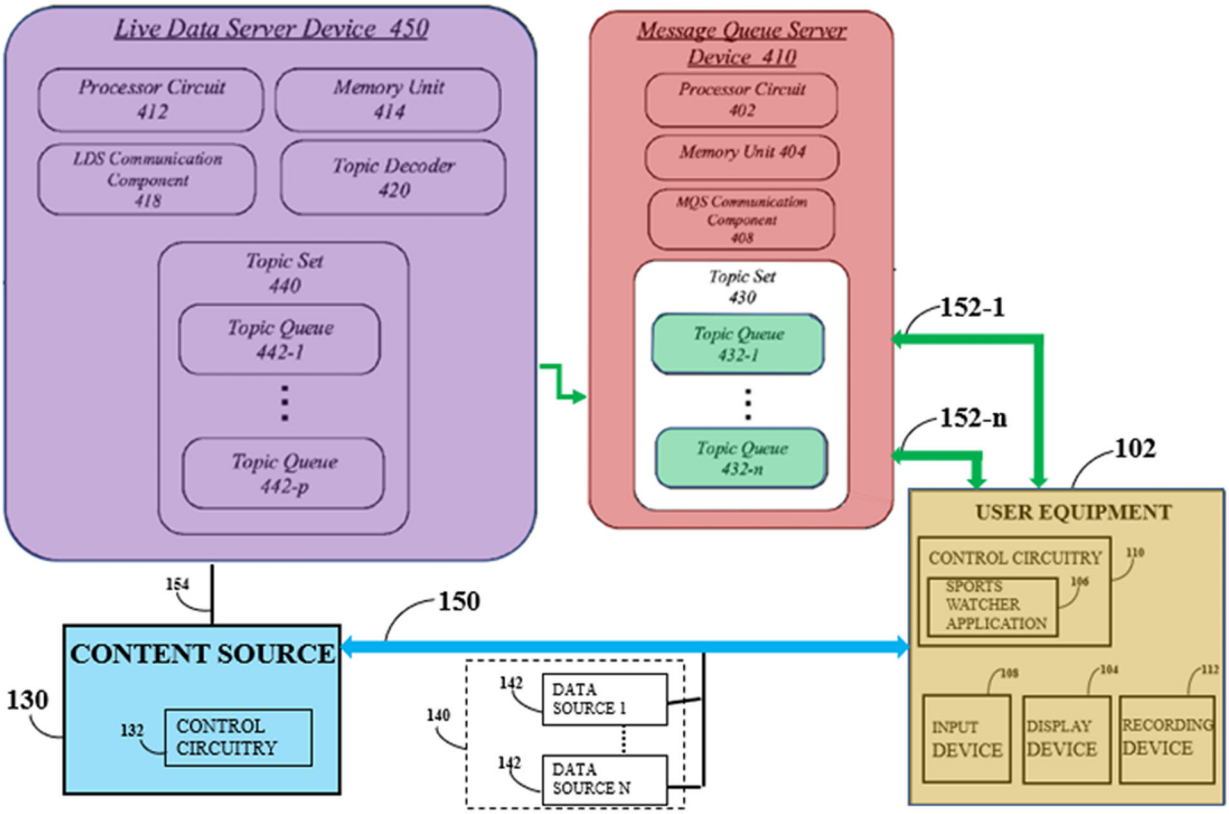
173. The image below illustrates Ellis's system, unaltered:



(EX-1016, Fig. 1.)

174. The next image below is one of Petitioner's and Dr. Wicker's making, which purports to replace Ellis's supplemental content source 120 and database 122 with an embodiment of Spivey's message queue server 120 that implements the

server as two separate components: message queue server device 410 and live data server device 450:

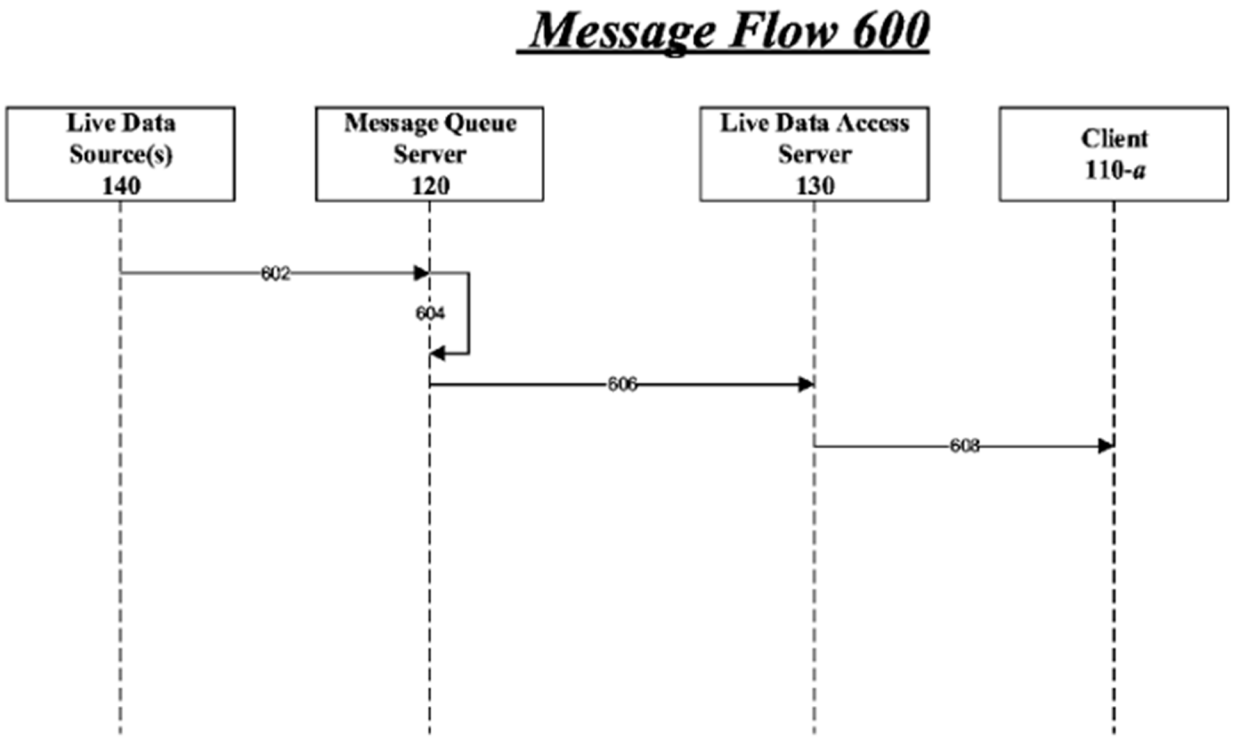


(EX-1001, ¶111.)

175. Rather than apply Spivey's teachings as they are, the proposed combination plucks one component (the message queue server) from the rest of Spivey's system and inserts it into a totally different system—thereby ignoring the very message flow that Spivey claims as inventive for reducing latency. Neither Petitioner nor Dr. Wicker explain how Spivey's message queue server would operate in the absence of Spivey's other components or how a POSITA would achieve Spivey's reduction in latency without the same message flow. Nor do Petitioner or

Dr. Wicker explain the appropriateness or rationale of selecting the message queue server (out of all of Spivey's disclosed components) for insertion into another system, much less a system such as Ellis that concerns video signals.

176. Spivey discloses a number of embodiments, all of which seek to minimize latency by relying on the same "message flow"—that is, a process in which data is moved from "live data sources" to the message queue server 120, from the message queue server 120 to the live data access server, and from the live data access server to the user (i.e., "client") devices:



(EX-1017, Fig. 6, ¶¶65-69.)

177. Spivey explains that:

> Message flow 600 begins when live data is received from live data source(s) 140 at message queue server 120 in message 602….Message

> queue server 120, upon receiving message 602, may sort the data payload of message 602 into a topic queue….In some embodiments, message queue server 120, e.g. using topic decoder 320, may format the data payload or otherwise alter the structure of the data payload….Once the data payload is inserted into a topic queue, message queue server 120 may push the data payload to live data access server 130 in message 606. Again, this assumes that live data access server 130 has subscribed to the topic queue selected in message 604. Upon receipt of message 606, live data access server 130 may push the data payload to client device 110-*a* in message 608, assuming both that client device 110-*a* selected the topic corresponding to the topic queue of message 604 and that the connection between client device 110-*a* and the live data access server 130 is still open.

(EX-1017, ¶¶66-69.)

178. Spivey consistently describes this same message flow throughout all embodiments of its invention. (*See, e.g.*, EX-1017, Figs. 1, 7, 11, 12, 14, 17, 22, ¶¶36-38, 46-49, 75-79, 91-100, 108, 126-132, 160.)

179. The portion of Spivey's system that Petitioner proposes implementing in Ellis is one embodiment <u>of the message queue server 120</u>. (EX-1017, ¶53.)

> Fig. 4 illustrates an embodiment of an operating environment 400. Operating environment 400 may illustrate a second embodiment ***of message queue server 120***. As shown in Fig. 4, ***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.

(EX-1017, ¶53 (emphasis added).) But this two-device architecture (message queue server device 410 and live data server device 450) is fundamentally an embodiment of the message queue server 120. Spivey teaches that this alternative embodiment of the message queue server 120 should be applied using the same message flow and system illustrated and described throughout its specification in order to achieve a reduction in latency. (*See, e.g.*, EX-1017, Figs. 1, 6, 7, 11, 12, 17, 22, ¶¶36-38, 46-49, 75-79, 91-100, (all referencing or illustrating "message queue server 120"); *see also id.* Figs. 14, 17, 22, ¶¶108, 126-132, 160 (all referencing "message queue server 1420," which is described as an "embodiment[] of message queue server 120" at paragraph 107).) Accordingly, the mere implementation of the message queue server 120 as "two separate devices" (FIG. 4) does not give license to impart new or different functionality to the message queue server 120 beyond the teachings of the reference; rather, Spivey demands that the same functionality of the message queue server 120 be consistently applied even when implemented using the two separate devices of FIG. 4.

180. Dr. Wicker does not support his opinion that Spivey contemplates a different message flow that would provide the same benefits. Dr. Wicker asserts that Spivey teaches a different flow, in which the message queue server might communicate directly with client devices, without a live data access server and without live data sources. (EX-1001, ¶99.) In his 500-page declaration, he barely

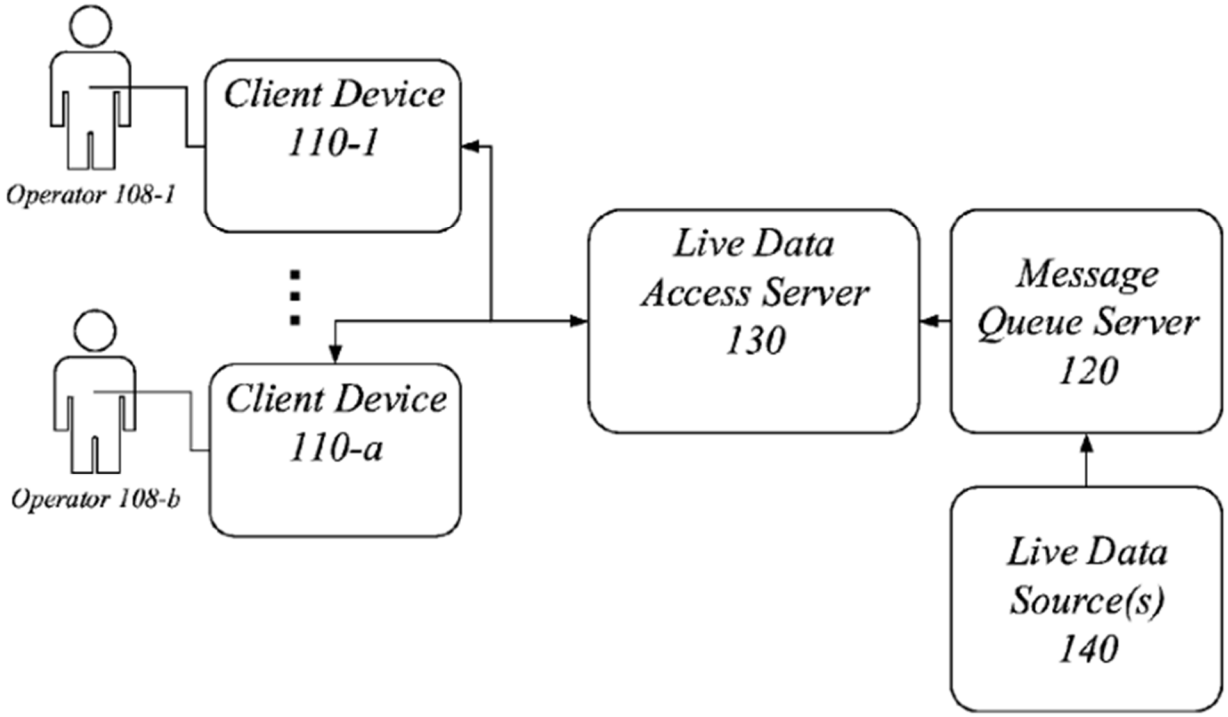
addresses this radical modification of Spivey's message flow, often repeating the same selective quote of Spivey's specification. (*See* EX-1001, ¶99, ¶562, n.3; *see also id.*, ¶597, n.9, n.13.) Specifically, Dr. Wicker states that "Spivey also discloses that the MQSD may be 'accessible to clients over the network directly.'" (EX-1001, ¶562, n.3; *see also id.*, ¶597, n.9, n.13.) But Dr. Wicker's quote leaves out the word "not." *Id*.

181. In fact, Spivey teaches that, in FIG. 1, the "message queue server 120 is ***not*** accessible to clients over the network directly." (EX-1017, ¶36.) Nowhere in the figures, specification, or claims, does Spivey describe an embodiment, in which the message queue server delivers data directly to client devices ***without*** the use of a live data access server. And nowhere does Spivey suggest that such an altered flow would achieve the same reduction in latency.

182. In reviewing Dr. Wicker's minimal discussion of this issue, he cites the following disclosures from Spivey: ¶¶36, 38, 46, 53, 84, Fig. 4. But none of these disclosures teach removal of the live data access server or disclose an embodiment lacking a live data access server.

183. Paragraphs 36, 38, and 46 all discuss "message queue server 120" or "message queue server 310," which is "a representative example of message queue server 120." (EX-1017, ¶¶ 36, 38, 46.) Message queue server 120 is illustrated in Figure 1 below, which shows the same message flow I describe above—i.e., from

live data source to message queue server 120, from message queue server 120 to live data access server, and from live data access server to client devices:



(EX-1017, Fig. 1.) While Spivey explains that the message queue server 120 "may process live data to provide live data quickly to the client devices 110-1 through 110-a," this statement does not indicate that it is the server that provides data ***directly*** to the client devices. (EX-1017, ¶36.) It merely conveys what it states—that the server processes live data.

184. Dr. Wicker places much emphasis on the fact that Spivey states the system "*may*" include a live data access server 130. (EX-1001, n.3 (quoting EX-1017, ¶38).) By that same logic, Spivey's system "may" include a message queue server or not, despite the fact that the message queue server is critical and necessary

to achieving Spivey's invention. (EX-1017, ¶¶35 ("The system **100** ***may*** include a message queue server **120**.").) The use of the word "may" in the context of Spivey does not, in fact, teach a POSITA how Spivey's system and alleged benefits would be achieved in the absence of a live data access server.

185. Dr. Wicker also relies on the ability of the message queue server to push data from topic queues to subscribers of the topics; however (and again), nothing in this stated ability indicates to a POSITA that the subscribers are necessarily client devices and that the client devices are receiving the data directly from the message queue server. (EX-1017, ¶46.) A POSITA would not understand Spivey to implement the server in this way, when Spivey consistently teaches that the live data access server receives client subscription requests, subscribes to the message queue server topics itself, and conveys data from the message queues to the subscribed client devices. (*See, e.g.*, EX-1017, Figs. 1, 7, 11, 12, 14, 17, 22, ¶¶36-38, 46-49, 75-79, 91-100, 108, 126-132, 160.)

186. As I have discussed above, the figure 4 embodiment and paragraph 53 merely describe an alternative embodiment of message queue server 120, in which it is implemented as two separate devices: a message queue server device 410 and a live data server device 450. (EX-1017, ¶53, Fig. 4.) Spivey explains that "[i]mplementing a message queue server as two separate devices may be useful to protect proprietary data that may be stored on live data server device 450 and/or to

limit potentially harmful access to the live data server device." (EX-1017, ¶57.) But nothing in this alternative embodiment of the message queue server teaches a POSITA to implement this embodiment outside of the message flow that Spivey consistently describes.

187. The final paragraph 84 that Dr. Wicker cites as support is no more illuminating. Spivey discloses a user interface that allows a client device to view and select topics available from a message queue server, to which the client wishes to subscribe. (EX-1017, ¶84.) Again, this is consistent with Spivey's repeated description of the live data access server subscribing on behalf of client devices to topics available at the message queue server. (*See, e.g.*, EX-1017, Figs. 1, 7, 11, 12, 14, 17, 22, ¶¶36-38, 46-49, 75-79, 91-100, 108, 126-132, 160.)

188. In what may be an attempted back-up argument that modifying Spivey's message flow would have been obvious, Dr. Wicker asserts in a footnote that a POSITA would have cut out the live data access server, because "reducing the amount of server transfers would reduce latency, which is Spivey's main purpose." (EX-1001, n.3.) However, this point from Dr. Wicker seems to be self-contradicting, given that Dr. Wicker's proposal for combining Ellis and Spivey includes adding two servers (Message Queue Server Device 410, Live Data Server Device 450) from Spivey to the system taught by Ellis—according to his own opinion, this combination increases latency. (EX-1001, ¶111.) Dr. Wicker fails to demonstrate

that the combination of Ellis and Spivey provides any advantages to the system described by Ellis that could motivate such combination, especially improvements in latency, which I further discuss below.

189. Dr. Wicker additionally fails to acknowledge the increase to latency when the live data access server is removed from Spivey's disclosed message flow. Spivey teaches that the live data access server provides a public interface between client devices and message queue server 120. (EX-1017, ¶38.)

> For example, live data access server 130 may allow operators 108 of client devices 110 to select topics related to a live event to subscribe to, and may retrieve the topic information and push the information to the client devices 110…. A client device, e.g. client device 110-1, may be usable by an operator, e.g. operator 108-1, to subscribe to and receive live data ***via live data access server 130***.

(EX-1017, ¶¶38-39 (emphasis added).) Spivey further teaches that by using the live data access server, data payloads can be immediately pushed to the subscribed live data access server, such that data payloads need no longer be stored on the message queue server. (EX-1017, ¶51.)

190. Assuming a POSITA were to reconfigure the message queue server to even operate without the assistance of the live data access server (which Dr. Wicker does not explain), these operations which Spivey allocates to the live data access server would have to be picked up by the message queue server 120, thereby slowing

down its operations and increasing latency. Rather than the message queue server's operations being focused to processing and pushing out data, the server would also need to manage all client subscriptions (as opposed to just subscriptions by the live data access server) and ensure that data payloads are conveyed from the correct message queues to the subscribed client devices. (*See* EX-1017, ¶¶66-69, 36-39.) This increased load on the message queue server would, in fact, increase latency and delay messages being received by the client devices.

191. Spivey repeatedly describes its reduction in latency (e.g., "at least two orders of magnitude shorter than conventional time periods") as a result of Spivey's message flow, including the live data access server. (*E.g.*, EX-1017, ¶96 (describing the flow of the data payload from live data access server to the subscribed client devices); *see also id.*, ¶¶69-70, 100-101.)

192. That is, Petitioner and Dr. Wicker do not merely apply Spivey's disclosure to Ellis's system. Rather, they remove the message queue server from the context of Spivey's invention and disclosed message flow, modify the message queue server to include additional responsibilities and capabilities, and do so without regard for the impact of these modifications and without explaining why a POSITA would have made such modifications in the first place. The resulting system suffers from many problems.

b. Petitioner's proposed combination of teachings suffers from many problems.

193. Even assuming one would use Spivey's message queue server while departing from all of Spivey's other teachings, the combination proposed by Petitioner and Dr. Wicker suffers from many problems.

194. ***First***, neither Petitioner nor Dr. Wicker have attempted to explain how the resulting system's message flow would operate or how a POSITA would have been able to predict its operation, since Spivey expressly teaches that the message queue server will push topic-specific data payloads onto its queue to be conveyed via the subscribed live data access server to user/client devices. (EX-1017, Fig. 6, ¶¶66-69.) As I have stated above, this message flow is critical to Spivey's stated goal to reduce latency and minimize delays. (Section VI(b)(1)(a), *supra*; EX-1017, ¶¶27-28, 95-96, Fig. 11.) In selectively plucking the message queue server from this message flow, and inserting it in a totally different system, Petitioner and Dr. Wicker do not explain how Spivey's alleged reduction in latency would be achieved. And a POSITA would not have been able to predict such a result or had a reasonable expectation of achieving such a result, when modifying Spivey's teachings.

195. ***Second***, in stripping Ellis of its supplemental content source and database, neither Petitioner nor Dr. Wicker explain: (1) from what source will the system receive the supplemental content (i.e., data) in question; and (2) how will Ellis's allegedly novel "interactive sports watcher application" be stored remotely?

196. Ellis describes the supplemental content source as "any suitable ***content source*** configured to provide real-time…sports-related information to user equipment." (EX-1016, ¶41 (emphasis added).) "Some or all of this sports-related information may be stored in database 122." (EX-1016, ¶42.) Spivey's live data access server is not described as originating or generating any content. Rather, when operating in Spivey's intended environment, the live data access server subscribes to the message queue server and acts as a conduit for transmitted data payloads from the message queue server to the client devices. (EX-1017, Fig. 6, ¶¶66-69.) In the Petitioner's proposed combination of teachings, neither Petitioner nor Dr. Wicker identify a source of supplemental content or sports-related information. Spivey teaches that the message queue server receives live data from one or more "live data source(s)." (*E.g.*, EX-1017, Figs. 1, 6, ¶66.) But by replacing the supplemental content source in Ellis with the message queue server, alone, Petitioner and Dr. Wicker do not explain from where the supplemental content would be obtained. A POSITA would not have a reasonable expectation of success in implementing the server in this way and could not predict the resulting operation.

197. In addition, removing the supplemental content source 120 from Ellis constitutes a reduction of Ellis's advantages. As an example, Ellis teaches that supplemental content source 120 may communicate with content source 130 for data transmission to user equipment 102. (EX-1016, ¶44.) The communications

between Spivey's supplemental content source 120 and content source 130 enables centralized data control and management on content source 130 and allows for load balancing and scalability needs. In addition, this streamlined workflow taught by Ellis ensures that security protocols can be consistently enforced on content source 130, with which the user equipment communicates.

198. Furthermore, Ellis discloses an allegedly novel "interactive sports watcher application" that is critical to "support[ing] the enhanced access and viewing functionality" Ellis sought to achieve. (EX-1016, ¶¶8-10, 336-9.) And while this application may operate partially on the client's device, the application is stored in the database of Ellis's supplemental content source and provided to the user from this source. (EX-1016, ¶44.) If a POSITA were to modify Ellis in the way Petitioner suggests—by replacing the supplemental content source—he or she would remove the database and, therefore, remove the ability to store the interactive sports watcher application, which is central to Ellis's invention, remote from the user equipment. (*See id.*) Many of the features Ellis discloses would not be possible without the interactive sports watcher application and flexibility afforded by storing said application remote from the user equipment. Accordingly, a POSITA would not have been motivated to make such a modification, nor would he have had a reasonable expectation of preserving the benefits of both references in the way Dr. Wicker describes.

199. ***Third***, Petitioner's proposed combination removes the scalability contemplated by Ellis's system and, therefore, the benefit Ellis sought to achieve. Ellis explains that its system contemplates numerous supplemental content sources. (EX-1016, ¶39.) While illustrating only one instance each of a content source and supplemental content source, Ellis explains that "in practice there may be numerous instances of content source 130 and supplemental content source 120" and that the image is only so limited for clarity. (EX-1016, ¶39.) This scalability is fundamental to Ellis's goal—improving upon conventional media applications that failed to efficiently manage and leverage the capabilities of multiple tuners in user equipment. (*See* EX-1016, ¶5.)

200. Neither Petitioner nor Dr. Wicker acknowledge this scalable aspect of Ellis's system, and neither specify whether Spivey's two servers would be used in place of *each* supplemental content server or replace *all* of the supplemental content servers. Indeed, because Petitioner has removed the supplemental content source 120 entirely from their recreated diagram, it is not possible to ascertain how Petitioner would plan to configure the Spivey servers in relation to the potentially numerous supplemental content sources contemplated by Ellis. Either way, the scalability of Ellis's invention and the benefit of leveraging multiple tuners in user equipment to tune into multiple supplemental content sources is lost.

201. ***Fourth***, Spivey does not disclose, contemplate, or relate to the delivery of live event video; therefore, a POSITA would not have found Spivey's teachings relevant to Ellis's television tuning system, nor would he or she have had a reasonable expectation of success in combining Ellis's and Spivey's teachings. Spivey fails to describe any technology relevant to, or to even use the term, "video" or "television." (*See generally* EX-1017.)

202. Indeed, video technology is not relevant to the technical problem Spivey attempts to address—i.e., sending live event data to a mobile phone faster by not saving such data to an intervening database. In describing the prior art, Spivey recognized that "[s]ome content providers, such as news and information services, provide data about live events to users who may not be present at the events." (EX-1017, ¶2.) Spivey defines the data payload as "one or more items of information about the live event in progress" and provides examples, "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." (EX-1017, ¶66.) Spivey recognized that "live event data content providers" using conventional systems faced difficulties in distributing "data about a live event," such as a sporting event. (EX-1017, ¶28.) For example, Spivey explains that these data providers struggled to efficiently distribute the information, due to the fact that received data must first be added to a database system, then

written to files and/or propagated to servers, before a client would then have to request that the new data be read from the server. (EX-1017, ¶28.)

203. Accordingly, Spivey discloses techniques to more efficiently communicate data or information about a live event by using a message queue server, capable of receiving, sorting, and pushing out data payloads "immediately," without interim storage of the data in a database. (EX-1017, Title, cl. 1, ¶¶51, 46, 128, 131; Section V(B), *supra*.)

204. Ellis, on the other hand, sought to leverage the fact that user televisions had evolved to include multiple digital or analog tuners, allowing for more than one channel or media content to be tuned, displayed, and/or recorded. (EX-1016, ¶5.) In particular, Ellis discusses the "idle time" associated with tuning and displaying a digital television channel, which must be demodulated and then decoded, demultiplexed, and then extracted, as well as possibly decrypted or further decoded—all taking an appreciable amount of time while displaying a blank or black television screen. (EX-1016, ¶6.)

205. In seeking to improve the television viewing experience, a POSITA would not have found Spivey's techniques for data transmission to mobile devices—or messaging queue technology in general—relevant.

206. Video streaming requires the continuous, ordered delivery of video and audio components for proper playback. Message queues—like those disclosed in

Spivey—are not designed for the high-frequency, continuous delivery needed for real-time streaming of video content; they are utilized for providing less frequent, reliable data transmissions. Given this, there is no teaching or disclosure that would inform a POSITA how to implement Spivey's message queue server alongside the television tuning techniques (and associated video delivery) disclosed in Ellis.

207. In addition, Spivey describes a process of placing data payloads into a topic queue, and then processing the queue with a "first-in first-out" sequence to push the data from the queue to a live data access server for viewer consumption. (EX-1017, ¶2.) However, the "first-in first-out queue" data queue, where data must wait for previous entries to be delivered before it can be processed, also adds latency (compared to data delivery that may not need to wait without said queue), which further makes message queues unsuitable for real-time video streaming systems. Because of this, it would not have been obvious to a POSITA at the time—without the use of hindsight—to implement Spivey's teachings alongside Ellis's delivery of video to television.

208. For example, FIG. 6 of Spivey clearly contemplates the delivery of "messages," via the coupled message queue server and live data access server; such discrete messages, while perhaps appropriate for "a unit of live data," would simply not be suited for the delivery of continuous streaming video content. Therefore, a POSITA considering Ellis—which is directed towards improving the delivery of

video to sports viewers by leveraging television tuning capabilities—would not have found Spivey relevant to Ellis's system. Nor would he or she have expected success in implementing Spivey's teachings within Ellis's video-based system.

209. While Ellis discloses the ability to tune into multiple video and data content sources, Ellis does not concern any type of coordination of the timing and receipt of video and data. Given the lack of disclosure in either Ellis or Spivey or any prior art instruction as to how to reconfigure the non-video related teachings of Spivey for operation with Ellis's enhanced television experience—e.g., such as coordinating the multiple feeds, a POSITA would not have predicted Petitioner's proposed result or expected to succeed. Indeed, Spivey's focus on immediately pushing data payloads out (even assuming the message flow issues above were resolved) would be incompatible with Ellis's video delivery environment, which contemplates leveraging multiple tuners to allow a sports viewer to flip back and forth between different video streams.

210. For example, Spivey's contemplated immediate pushes of data payloads are inherently not gated, throttled or otherwise controlled. The very purpose of Spivey's data delivery is to transmit said data immediately. This teaching is incompatible with video delivery because absent some form of controlling the delivery of said data (e.g., the speed, timing or other parameter of the data), such data would not be delivered in a manner comprehensible with another, separately

delivered medium, e.g., video. In this regard, absent some controlling mechanism for the provision of said data (which Spivey does not teach), a POSITA would understand that the immediate "push" provision of data would be incompatible with Ellis because such data would not be capable of delivery in a manner that matches the video content.

211. Dr. Wicker's argument suggests that any architecture could be slotted into Ellis's interactive media system, merely because Ellis does not specify the detailed architecture of the supplemental content source 120. (EX-1001, ¶107.) But as detailed above, there are many significant drawbacks to Petitioner's proposed implementation of a modified version of Spivey within Ellis.

212. Accordingly, for the reasons stated above, implementing a modified portion of Spivey's system within Ellis's system would not have been a simple substitution of known elements to obtain predictable results. (*See* EX-1001, ¶114.) The altered operation of Spivey was not known. And removing the supplemental content source and its database from Ellis in favor of a couple components from Spivey would not have been a simple substitution, as doing so would have decreased the functionality of the Ellis system, left the system without a source of supplemental content, and rendered many of its intended benefits moot.

c. A POSITA would not have been motivated to combine the references' teachings.

213. In light of the many problems with Petitioner's proposed combination of teachings, which I discuss above, a POSITA would not have been motivated to modify and combine the references' teachings. (*See* Section VI(A)(1)(a), *supra*.) Dr. Wicker makes a number of assertions to the contrary, each of which, I address below.

214. ***First***, Dr. Wicker's primary reason that a POSITA would have allegedly implemented Spivey's teachings within Ellis's system is to minimize latency. (EX-1001, ¶¶108-110.) Specifically, Dr. Wicker asserts a POSITA "looking to improve Ellis's latency issues—would have been drawn to a system such as Spivey" in order to "achieve full-duplex communications between a client and a remote host while minimizing latency." (EX-1001, ¶108.) There are many errors in this reasoning.

215. Ellis does not disclose any issues with latency or delay, and neither Petitioner nor Dr. Wicker explain why Ellis's system would have been understood to suffer from such issues or the cause of such issues. Dr. Wicker refers to "Ellis's latency issues" without actually explaining why or how a POSITA would have understood Ellis to suffer from latency issues. (EX-1001, ¶108.) While the Ellis reference claims priority on its face back to 2006, the application that resulted in the Ellis reference was filed in 2014, several months before the application for Spivey was filed. (EX-1016, p.1; EX-1017, p.1.) Dr. Wicker claims that applying the

limited, modified teachings of Spivey would achieve transmission to client devices in a period of time that "is at least two orders of magnitude shorter than conventional time periods of legacy devices." (EX-1001, ¶110 (quoting EX-1017, ¶96).) But Ellis is not a legacy device, nor has Dr. Wicker shown it to be.

216. Furthermore, all of Spivey's disclosures of a "full-duplex connection, such as a websocket connection" that purportedly minimized delay, describe this type of connection as occurring between the live data access server 130 and the client devices. (EX-1017, ¶¶43, 61, 74, 92.) But Petitioner's proposed combination of teachings ignores the live data access server, instead incorporating only the message queue server. While this message queue server includes a live data server device, Spivey describes this device as being on the non-public, non-client side (EX-1017, ¶57), which is also how Petitioner has proposed use of the 2-component message queue server (Petition, 19). That is, applying Spivey's limited, modified teaching in the way that Petitioner suggests does not include the "full-duplex connection" Spivey discloses.

217. For this reason, and the reasons I have stated above regarding altered message flow and the lack of a live data access server (Section VI(A)(1)(a)-(b)), neither Petitioner nor Dr. Wicker has shown how the proposed combination of teachings would actually minimize latency.

218. Dr. Wicker additionally goes on to restate the arguments made in the petition, in nearly a word-for-word fashion, although the paragraphs have been rearranged. (*Compare* EX-1001, ¶¶111, 113-115, *with* Petition, pp. 21-24.) Specifically, Dr. Wicker asserts that a POSITA would have been 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-1001, ¶115.) Dr. Wicker cites only the abstract of Spivey and nothing else; he does not explain what he means by "complementary solution" or to what problem Spivey is allegedly a complementary solution. (*See id.*)

219. As I have discussed above, because Dr. Wicker modifies the teachings of Spivey by removing the message queue server from the message flow Spivey discloses, a POSITA would not have been able to predict the resulting "solution" or how such a "solution" would have operated. (Sections VI(A)(1)(a)-(b), *supra*.) Thus, a POSITA would not have recognized Ellis and Spivey to disclose "complementary" means of achieving a common goal.

220. Additionally, Dr. Wicker asserts that a POSTA would have been motivated to modify Ellis in view of Spivey, because both references related to "content delivery for real-time events." (EX-1001, ¶113.)

221. However, as I have explained previously, Spivey is not analogous to video delivery systems and contains no disclosure relevant to video delivery

systems. (Section VI(A)(1)(b), *supra*.) Furthermore, I am informed that the fact that references are drawn from the same broadly characterized field of is insufficient evidence, alone, of a POSITA's motivation to combine.

222. Neither Petitioner nor Dr. Wicker have explained why or how a POSITA would have removed the message queue server from the allegedly novel message flow Spivey touts or how any of the benefits taught by Spivey would have been preserved in the resulting system. Indeed, as I have explained in detail, a POSITA attempting to combine the teachings identified by Petitioner would render Ellis's system non-operational and lose the benefits of both Ellis's and Spivey's individual teachings.

223. Accordingly, a POSITA would not have been motivated to attempt such a combination of teachings.

2. **The asserted combination does not disclose or render obvious an "*event socket*" and/or *means for transmitting the first score information therethrough* or transmitting score information "*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.*"**

224. The independent claims recite the following limitations related to an "*event socket*":

Claim Limitation	Claim Language
19.10	***transmit at least the first score information*** *the at least some of the first event information to the first viewer client device*

Claim Limitation	Claim Language
	of the first plurality of viewer client devices ***via a third 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; and*
23.3	***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.* *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.*
27.3	*means for transmitting the first score information to at least the first viewer client device via a first event information communication channel that is different from the first video communication channel.*

225. As I explained above, a POSITA would understand the claimed "means for transmitting" to be a socket server transmitting through an event socket that implements one or more of the algorithms described in Figures 2, 3, 21A-21E and accompanying disclosures, including 5:29-44 9:16-10:43, 19:64-20:61, 21:37-22:67, 23:20-45, 26:50-27:27, 29:25-43, 46:61-47:11, 51:27-49, and 53:21-37 of the '697 patent. (*See* Section IV(B)(1), *supra*.)

226. Petitioner and Dr. Wicker identify Spivey's message queue server device (MQSD) as the alleged socket server. (Petition, 35; EX-1001, ¶591.)

227. Petitioner and Dr. Wicker allege that Spivey's topic queues disclose the '697 patent's first event sockets and further allege in a footnote that "a POSA would have understood Spivey to disclose a direct connection between the client devices and MQSD." (EX-1001, ¶¶596-597, 616, 628; Petition, 38-40, 49-50, 56, n.8.)

228. It is my opinion that the asserted combination of teachings does not disclose or render obvious limitations 19.10, 23.3, and 27.3, because: (1) Spivey does not, as Petitioner alleges, disclose a "websocket connection" or channel between the message queue server (comprising the message queue server device and live data server device) and a viewer client device, nor would a POSITA have been motivated to modify the prior art teachings to achieve such a channel; (2) the Petition fails to show that Spivey's topic queue are "event sockets" under Petitioner's

implicit construction; and (3) a POSITA would not have been motivated to combine or modify the references in such a way as to teach the claimed limitations.

a. The asserted combination does not disclose or render obvious transmitting score information via a communication channel between an event socket and a viewer client device

229. Independent claims 19 and 23 require that the socket server transmit score information via a communication channel ***between*** ***an event socket and a viewer client device***. (EX-1006, cl. 1.) Relatedly, claim 27 requires a means for transmitting the score information—i.e., a socket server transmitting through an event socket. (*See* Section IV(B)(1), *supra*.)

230. Petitioner's arguments as to this limitation hinges on its implementation of Spivey's teachings within Ellis—specifically, Petitioner and Dr. Wicker assert that a POSITA would have implemented Spivey's message queue server (comprising the message queue server device and live data server device) in place of Ellis's supplemental content source. (EX-1001, ¶¶596-602 616, 628.) In particular, Dr. Wicker alleges that Spivey discloses a "websocket connection" between the message queue server device's topic queue (the alleged "*event socket*") and client devices, allegedly establishing a "*channel*" between the message queue server device's topic queue and the viewer device. (EX-1001, ¶¶601, 616, 628.)

231. Spivey only discloses a "websocket connection" as occurring between the client devices and the live data access server 130—i.e., a server which Petitioner

and Dr. Wicker have not included in the proposed combination of teachings. (EX-1017, ¶¶43, 61, 74, 92; *see also* Section VI(A)(1), *supra*.) Spivey only discloses the "live data connection" between the client devices and the live data access server, which is not included in Petitioner's proposed combination of teachings. (Sections VI(A)(3)(a), VI(A)(1), *supra*.) Spivey does not disclose a connection between the message queue server device and client devices and would not, therefore motivate or teach a POSITA to implement the message queue server device in the way that Petitioner proposes. (Sections VI(A)(1)(a), (c), *supra*.)

232. Dr. Wicker selectively quotes Spivey to suggest that "client devices 'request a live data connection' ***with the MQSD [message queue server device]*** by 'select[ing] or enter[ing] the address or connection information ***to a particular' MQSD***." (EX-1001, ¶596 (quoting EX-1017, ¶¶81-82) (emphasis added); *see also id.* ¶602.)

233. In fact, Spivey teaches:

> UI 800 may include various UI elements that allow an operator 108 of a client device 110 ***to request a live data connection with <u>a live data server</u>***…. UI 800 may also include an address UI element 830 to allow the operator to select or enter the address or connection information ***to a particular <u>live data access server</u>***…. In an embodiment, address UI element 830 may be a text field or a pull-down menu pre-populated with addresses ***of live data access servers*** available to live data app 210.

(EX-1017, ¶¶81-82.) That is, even if Spivey's teachings are applied within Ellis, Spivey does not disclose the identified websocket connection between the message queue server (message queue server device) and the client devices.

234. Dr. Wicker also asserts that Spivey teaches a direct connection based on the following quote from Spivey, that "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." (EX-1001, ¶597 (quoting EX-1017, ¶46).) But in this very embodiment which Dr. Wicker quotes, Spivey teaches that "MQS communication component 308 may include hardware and/or software that allows MQS device 310 to transmit and receive electronic data, for example live data, with other devices, ***such as live data access server 130***." (EX-1017, ¶46.) The statement that data is pushed from topic queues to "subscribers" is merely a statement of how the message queue server's topic queue functions. (*See id.*) But a POSITA would not understand this simple statement to direct a POSITA away from the message flow that Spivey consistently teaches throughout its specification, which uses the live data access server to subscribe to topic queues on behalf of client devices:

- "In response to receiving the selected topic(s) in message 718, live data access server 130 may then subscribe to the selected topic(s) with message 720 to message queue server 120. Subscribing to a topic may include registering with message queue server 120 that live data access server 130 is to receive any data payload placed in a topic queue for that topic." (EX-1017, ¶¶78-79.)

- "For example, live data access server 130 may allow operators 108 of client devices 110 to select topics related to a live event to subscribe to, and may retrieve the topic information and push the information to the client devices 110." (EX-1017, ¶38.)

- "A client device, e.g. client device 110-1, may be usable by an operator, e.g. operator 108-1, to subscribe to and receive live data via live data access server 130." (*Id.*, ¶39.)

- "MQS communication component 308 may then push the data payload from the topic queue to a live data access server that has subscribed to the topic of the topic queue." (*Id.* ¶49.)

- "A topic queue 332 may be a queue data structure, in particular a first-in first-out queue, where data payloads are placed and immediately pushed to a subscribing live data access server. In an embodiment, the data payloads may not be stored on MQS device 310 once pushed out to the subscribing live data access server." (*Id.*, ¶51.)

- "Subscription component 540 [of the live data access server 510] may manage subscriptions to topics for currently connected clients." (EX-1017, ¶64.)

- "Of note is that the message flow 600 assumes that client device 110-a has already established an open connection with live data access server 130 and has selected at least one topic to subscribe to, and that live data access server 130 has already subscribed to the at least one selected topic at the message queue server 120." (EX-1017, ¶65.)

- "Again, this assumes that live data access server 130 has subscribed to the topic queue selected in message 604." (EX-1017, ¶¶68.)

- "LDA [live data access] communication component 520 may also operate to receive connection requests from client devices 110 and to authenticate the requesting client. In some embodiments, operators 108 may need to register with live data access server 510 in order to be able to subscribe to and receive the live data." (EX-1017, ¶62.)

- "Having saved favorites may allow live data access server 130 to subscribe automatically to the saved topics whenever a connection is opened with that client." (EX-1017, ¶87.)

- "The logic flow 1100 may push the data payload of the topic entry to the client device via the connection at block 1110. For example, live data access server 130 may push the data payload to client device 110, and to any other connected client devices that also requested that topic." (EX-1017, ¶96.).

235. Accordingly, neither Petitioner nor Dr. Wicker has shown that the proposed combination of teachings—lacking the live data access server—would result in a websocket connection between the message queue server device's topic queue and client devices. That is, neither the Petition nor Dr. Wicker show or explain how Spivey teaches how clients could connect to the topic queues, and therefore implementing Spivey's teachings in the way they propose does not result in the

alleged "communication channel ***between*** ***an event socket and a viewer client device.***"

236. Neither Petitioner nor Dr. Wicker allege that a POSITA would have been motivated to modify Spivey's teachings on this point and neither acknowledge the significant drawbacks associated with doing so. (*See* Section VI(A)(1), *supra*.) Dr. Wicker merely notes in a single sentence that a POSITA "would have used this implementation [which is not actually taught by Spivey] because reducing the amount of server transfers would reduce latency." (EX-1001, ¶597.) But as I have explained in detail above, Dr. Wicker's proposed replacement of Ellis's supplemental content server with the message queue server device and live data server device, in fact, increases the number of servers being used in the resulting system, thereby increasing latency. (Section VI(A)(1), *supra*.) Furthermore, omission of the live data access server Spivey relies on for its message flow would further contribute to increased latency. (*Id.*)

237. To the extent Dr. Wicker simply relies on his generalized discussion of motivation to combine, nothing in his discussion explains why or how a POSITA would have modified the message flow Spivey discloses for reducing latency or the impact of such modifications. Furthermore, I have addressed the flaws in Petitioner's and Dr. Wicker's generalized motivation to combine the teachings of Ellis and Spivey, above. (Section VI(A)(1), *supra*.) A POSITA would not have been

motivated to modify Spivey's teachings and apply them to Ellis when doing so would render Ellis's system non-operational and lose the benefits of both Ellis's and Spivey's individual teachings. (Section VI(A)(1)(c), *supra*.)

238. Additionally, and as further reason this limitation is not met, I note that Dr. Wicker relies on Ellis's teaching that the supplemental content server may be configured to provide "sports scores and game summary information" to users as allegedly disclosing transmission of score information. (EX-1001, ¶¶600, 616, 628.) However, according to his proposed combination of teachings, Ellis's supplemental content source would no longer exist. (*See* EX-1001, ¶111.) Dr. Wicker does not address this inconsistency or elaborate on how or why such a teaching might have been applied in the combination of prior art teachings.

239. Accordingly, it is my opinion that the cited combination of prior art teachings fails to disclose or render obvious transmission of score information to a viewer client device via a communication ***channel between an event socket and the viewer device***.

b. Petitioner Fails to show that Spivey's "topic queues" are "*event sockets*."

240. The Petition fails to show that Spivey's topic queues are "*event* ***sockets***" under Dr. Wicker's and the Petition's implied definition that requires them to be "uniquely addressable communication endpoints . . . ." Spivey's topic queues are merely an internal data structure to the message que server. The Petition does not

identify any disclosure that would suggest to a POSITA that they are communication endpoints, or how they are uniquely addressed with Spivey. To the contrary, Spivey's disclosures demonstrate the opposite, as I explained below. Therefore, in my opinion, the Petition fails to show that Spivey's message queue server device meets the limitations of claim 1 of the '697 patent, regardless of whether said message queue server communicates with any client devices as the Petition alleges. Petitioner also alleges that "a POSA would have understood that Spivey's topic queues disclose first/second event sockets." (Petition, 41 (emphasis added).) However, as described in Spivey, the topic queues are not "uniquely addressable communication endpoints," per Spivey. (EX-1017, ¶¶46-54.) Instead, Spivey discloses that the topic queues are internal to the MQS, and Spivey does not disclose that they are "uniquely addressable communication endpoints." In my opinion, a POSA would have understood that Spivey's "topic queue" is merely an organization of data, such as a database table, that includes information about a particular topic. For example:

> [0051] A topic queue 332 may be a ***queue data structure***, in particular a first-in first-out queue, where data payloads are placed and immediately pushed to a subscribing live data access server. In an embodiment, the data payloads may not be stored on MQS device 310 once pushed out to the subscribing live data access server.

(EX-1017, ¶51.)

241. Although Spivey does not provide details of the structure of these topic queues, a POSA would not consider a database, message queue, or any other internal data structure by themselves to be a "uniquely addressable communication end-points." A POSA would similarly not consider the topic queues, as disclosed in Spivey, to be uniquely addressable because the queues themselves lack any address through which they communicate at all. (EX-1017, ¶51). Spivey instead states that a message queue server (MQS) communication component 308 "may push data from topic queues," but does not provide the details about that process, including any indication that any of the queues themselves are "uniquely addressable communication endpoints" in the system:

> [0046] Message queue server 310 may include a message queue server (MQS) communication component 308. MQS communication component 308 may include hardware and/or software that allows MQS device 310 to transmit and receive electronic data, for example live data, with other devices, such as live data access server 130. In an embodiment, MQS communication component 308 may also receive requests to subscribe to topics in a topic set 330, and may push data from topic queues 332-1, 332-n to the subscribers of the topics corresponding to those topic queues.

(EX-1017, ¶46.) And without more, a POSITA would not consider such a database, message queue, or other internal data structure by itself to be a "uniquely addressable

communication endpoint" because as disclosed by Spivey, the queues themselves lack any address through which they communicate at all. The only potential "communication endpoint" Spivey discloses as being "uniquely addressable" is the live data access server, not any of the topic queues at the message queue servers. (EX-1017, ¶¶38-39.)

242. For example, Petitioner relies on the following portion of the '697 patent specification for its definition of "event sockets": "The first event socket has a corresponding EventID that the VCD 'uses … to connect to the … socket,' e.g., by using a 'URL including the … EventID in a path of the URL.'" (Petition, 39 (citing EX-1006, 21:53-65); Ex-1001, ¶561) But the only URL that the Spivey clients are provided is for the live data access server, not the message queue server and certainly not the topic queues therein: "UI 800 may also include an address UI element 830 to allow the operator to select or enter the address or connection information to a particular live data access server." (EX-1017, ¶82 ("The address may resemble a uniform resource locator (URL), for example, 'torq.cbssports.com/torq.' In an embodiment, address UI element 830 may be a text field or a pull-down menu pre-populated with addresses of live data access servers available to live data app 210.").) Nowhere does Spivey provide any topic queue with its own "unique" URL—and neither Petitioner nor Dr. Wicker argue otherwise.

243. This architecture is fundamental to how the system described in Spivey operates: the client connects to the live data access server and selects the topics they subscribe to; the live data access server then subscribes to those topics on the MQS for the client. (EX-1017, ¶¶49, 128, 131) When there is an update for a topic queue, the live data access server must receive that update, determine which of the clients are subscribed to that topic, and then push that data to the client. (EX-1017, ¶¶38, 46, 49.) There is no disclosure of any connection for the client to anything other than the live data access server, and certainly no disclosure that each topic queue is a "uniquely addressable communication endpoint." (*See generally* EX-1017.)

244. Moreover, the URL disclosed by Spivey, which is "torq.cbssports.com/torq" does not uniquely address any topic queue at all. (EX-1017, ¶ 82). To the contrary, that URL identifies an overarching live data access server: "UI **800** may also include an address UI element **830** to allow the operator to select or enter an address or connection information to a ***particular live data access server.***" (Ex-1017, ¶ 82).

245. Each live data access server, in turn, can subsume ***many topic queues*** within it: "[C]lient device **200** may optionally request a ***topic list*** from live data access server **210** in message **708**. For example, the live data app **210** may request the topic list in order to be able to display a selection user interface element to allow the operator to ***select one or more topics to subscribe to***." (EX-1017, ¶ 75).

246. Therefore, the URL in Spivey that Dr. Wicker points to does not ***uniquely address*** any given topic queue.

247. Consequently, the Petition fails to show how Spivey's topic queue satisfy Dr. Wicker's construction of "event socket," which is, "a ***uniquely addressable*** communication endpoint on a server where first event information is provided to a VCD." (EX-1017, ¶75; EX-1001, ¶ 561).

248. Moreover, Spivey explicitly teaches—and a POSITA would understand—that a topic queue is a "data structure." (EX-1017, ¶51.) Other examples of data structures include stacks, graphs, hash tables, trees, or lists.

249. A POSITA would not understand a data structure, without more, to be the same as a socket or an endpoint. (*See* Sections III(A)(2)-(3), *supra*.)

250. A topic queue as disclosed in Spivey without more—like other types of data structures—would not be understood to be a "uniquely addressable communication endpoint."

251. Petitioner and Dr. Wicker attempt to muddy the waters because Spivey discloses that the topic queues may be identified by text strings. Petition at 40-41 ("Client devices use these topic label text strings to connect and subscribe to MQSD's topic queues.") They cite to Spivey Figure 9, which shows text strings for Topic Queues but notably are ***not*** URLs themselves:

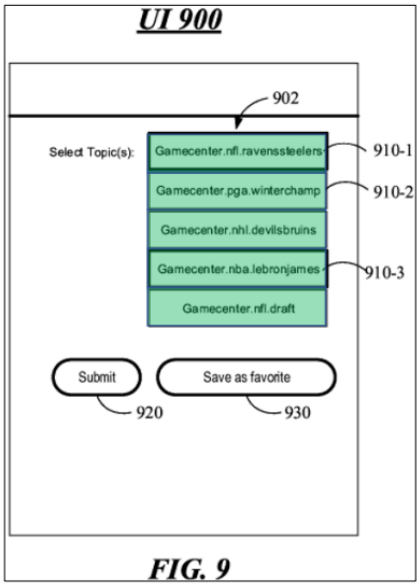


FIG. 9

(Petition, 41.)

252. However, even under Dr. Wicker's definition that requires a "socket" to be a "uniquely addressable communication endpoint on a server where first event information is provided to a VCD," the existence of a "text identifier" for each "topic queue" does not mean that they are either "communication endpoints" or "uniquely addressable." For example, Spivey does not disclose that any URL in its system includes the topic text identifier or that a URL corresponds to any specific topic queues. The only components that have an address corresponding to a URL in Spivey are the servers (specifically the live data access server not the message queue server) themselves, not the topic queues within the message queue server. Indeed, Dr. Wicker incorrectly states that the client devices disclosed in Spivey "use…topic label text strings to connect and subscribe to MQSD's topic queues." (Ex- 1001, ¶564). They do not—Spivey does not disclose that the text stings are used to

"connect" to the MQS's topic queues. To support this incorrect statement, Dr. Wicker then incorrectly states that the client devices "request a live data connection" with the MQSD by "selecting or entering address or connection information to a particular MQSD," citing paragraphs 81-82 of Spivey in support thereof. (Ex-1001, ¶564). But those paragraphs of Spivey explicitly state the client devices request a live data connection with the live data access server, not the message queue server (and not the topic queues within the message queue server):

> UI 800 may include various UI elements that allow an operator 108 of a client device 110 ***to request a live data connection with a live data server***…. UI 800 may also include an address UI element 830 to allow the operator to select or enter the address or connection information ***to a particular live data access server***…. In an embodiment, address UI element 830 may be a text field or a pull-down menu pre-populated with addresses ***of live data access servers*** available to live data app 210.

(EX-1017, ¶¶81-82.)

B. Ground 2 Does Not Disclose Or Render Obvious All Limitations of Claims 19, 23, and 27

253. I understand that Petitioner's Ground 2 relies on Herzog in view of Spivey and Ellis. (Petition, 60; EX-1001, ¶165.)

Ground 2	§103	Herzog in view of Spivey and Ellis	Claims 19-25, 27-30

254. Specifically, Petitioner and Dr. Wicker assert that a POSITA would have implemented Spivey's disclosure of a message queue server device in Herzog's system, and then modified the resulting system based on the teachings of Ellis. (Petition, 58-62; EX-1001, ¶¶142-163.)

255. First, a POSITA would not have been motivated to make such a combination when doing so would render the resulting system non-functional and moot the very benefits that each reference sought to achieve. Furthermore, even if the teachings of Ellis and Spivey are combined as Petitioner suggests, the resulting system would not disclose or render obvious all limitations of claims 19, 23, and 27.

256. Accordingly, Ground 2 does not disclose or render obvious all limitations of claims 19, 23, and 27.

1. A POSITA would not have been motivated to combine the teachings of Herzog and Spivey

257. Petitioner's proposed combination of Herzog's and Spivey's teachings closely mimics its proposed combination under Ground 1. Petitioner and Dr. Wicker assert that a POSITA would have been motivated to "incorporat[e] Spivey into Herzog's current data delivery system by implementing Spivey's MQSD [message queue server device] at Herzog's Origin Server to achieve a predictable result of decreased latency data delivery." (Petition, 60; EX-1001, ¶128.) The distinction from Ground 1 is that the message queue server device—i.e., one of the two components disclosed as forming the message queue server—has been implemented

alone. As their purported reasons for this combination mirror those reasons put forth under Ground 1, my response under Ground 2 is very similar.

258. That is, a POSITA would not have been motivated to combine Herzog's and Spivey's teachings, because: (1) Petitioner's proposed combination radically modifies the teachings of Spivey; (2) the resulting combination of Herzog's and Spivey's teachings suffers from many technical problems; and (3) a POSITA would not have been motivated to make such a combination when it renders the resulting system non-operational and lacking the benefits sought by each reference.

259. As an additional preliminary matter, the majority of the petition and Dr. Wicker's declaration rely on the proposed combination I describe above. But a single sentence included in the petition suggests an alternative, unspecified combination that potentially includes more of Spivey's system: "To a POSA, it would have been obvious to substitute any of Herzog's metadata 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." (Petition, 65.) As neither Petitioner nor Dr. Wicker articulate any specific combination of teachings other than the one I describe above, and neither support this statement that a POSITA would have substituted "any of Herzog's…elements with any combination of Spivey's" elements, I will address the specific combination proposed in their limitation-by-limitation analysis.

a. Petitioner's proposed combination radically modifies Spivey's teachings.

260. Petitioner and Dr. Wicker argue that a POSITA would have implemented a single component of Spivey's system—i.e., the message queue server device, in place of Herzog's origin server. (Petition, 60-65; EX-1001, ¶128.) But Petitioner fails to articulate a way in which these cherry-picked disclosures could be combined into an operational system. Even if one were to assume that such disclosures could be combined into an operational system, which is not taught by the prior art, the resulting system would lack the very benefits that Herzog and Spivey independently sought to achieve.

261. The figure below illustrates "system 40" contemplated by Herzog, unaltered, in which mobile device 45 receives video and audio data and telemetry data (e.g., metadata) from a content provider 42:

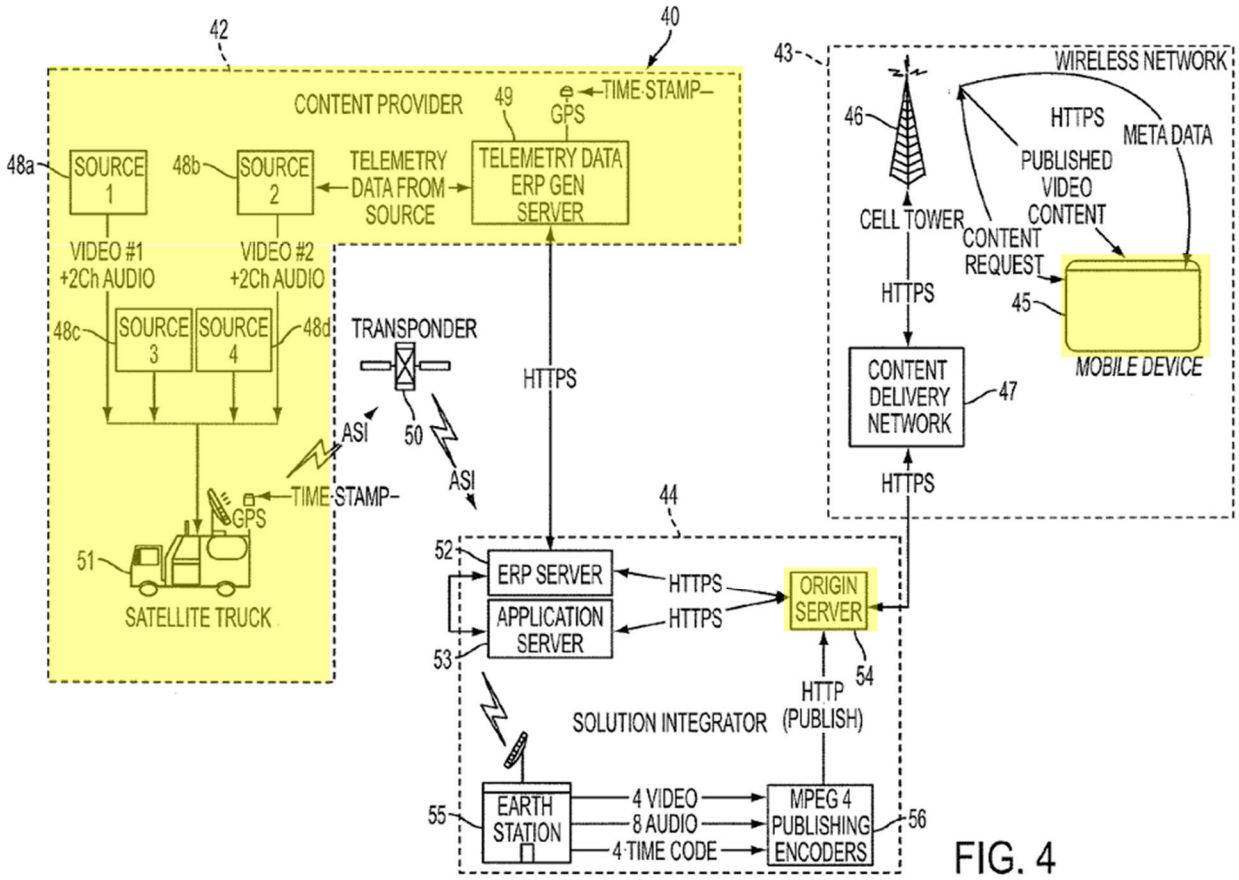
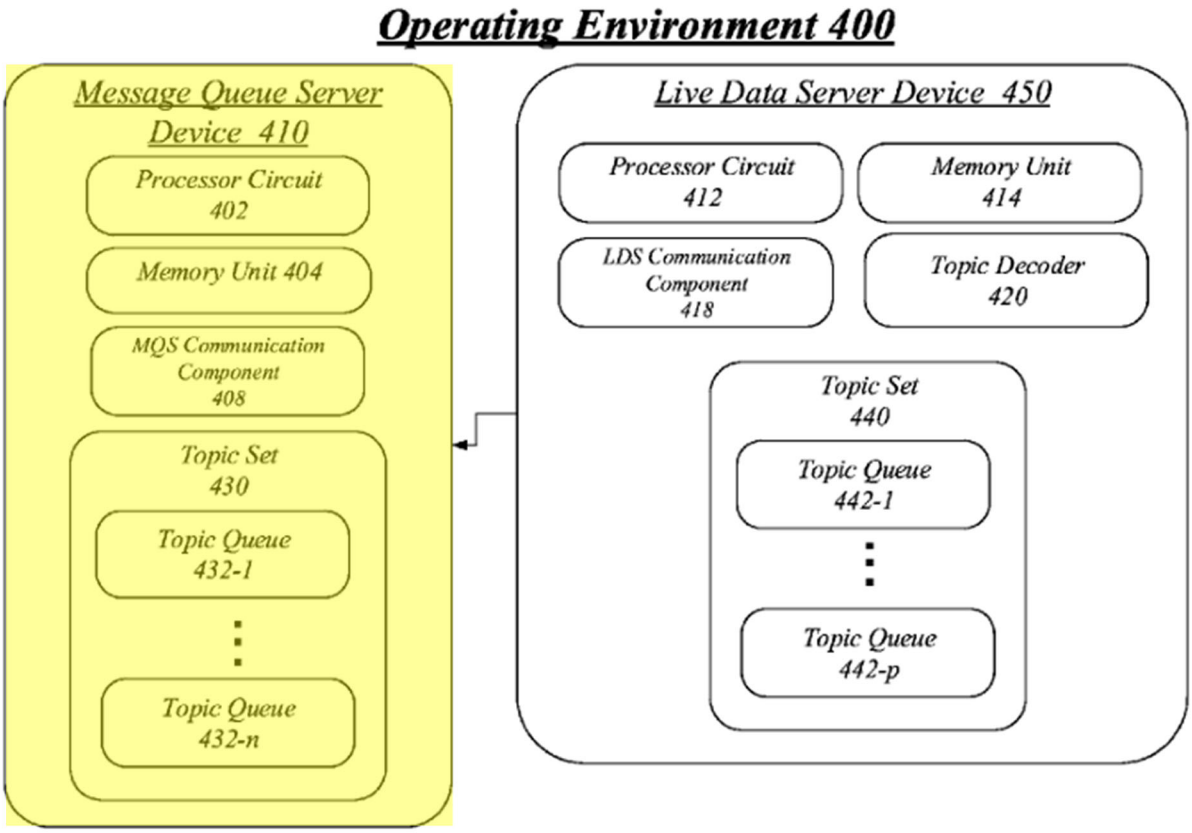


FIG. 4

(EX-1015, Fig. 4 (highlighted), ¶¶43-45.) System 40 is described as including a solution integrator 44, which in turn, includes an ERP server 52, an Application server 53, earth station 55, encoders 56, and—relevant to my analysis here—an origin server 54. (EX-1015, ¶48.)

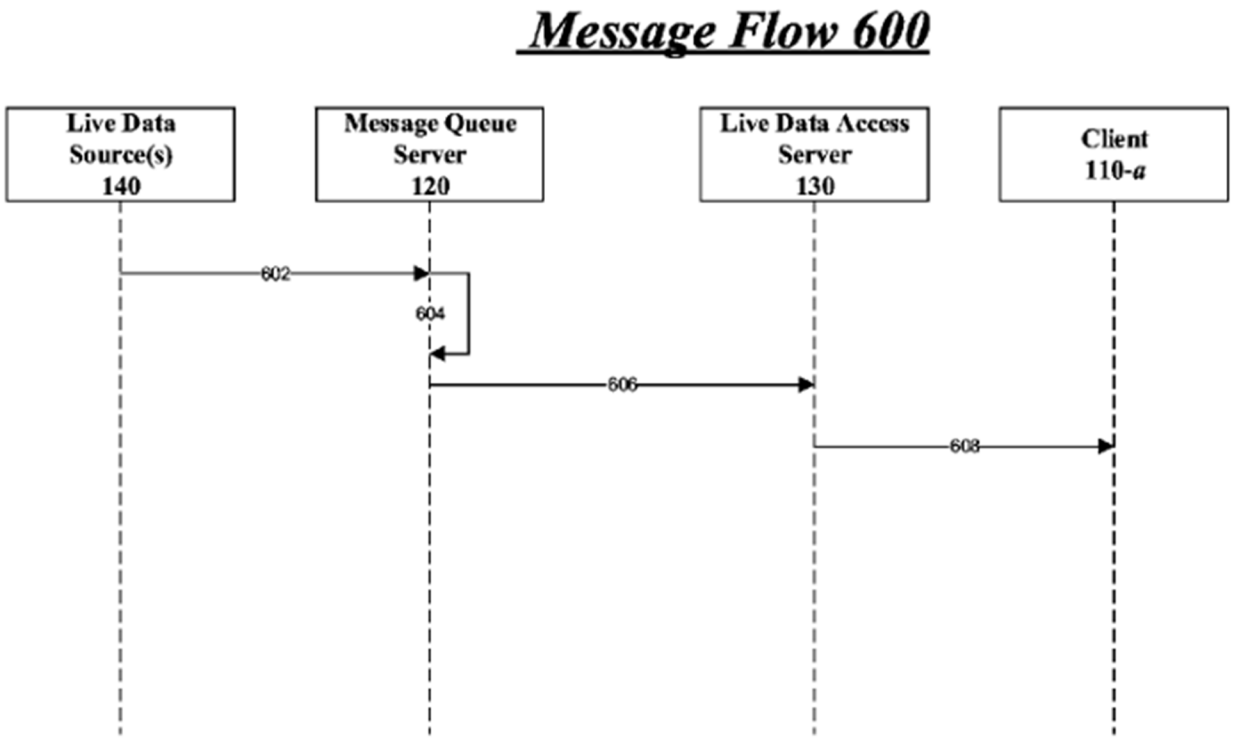
262. Petitioner and Dr. Wicker purport to replace Herzog's origin server with one component of one embodiment of Spivey's message queue server 120. (Petition, 62-63; EX-1001, ¶143.) Figure 4 of Spivey, below, illustrates a single embodiment of message queue server 120, which implements the server as two individual components: message queue server device 410 and live data server device 450:



(EX-1017, Fig. 4 (highlighted), ¶53.)

263. However, rather than apply Spivey's teachings as they are, the proposed combination removes one component (the message queue server device 410) from the rest of Spivey's system—and therefore does not fully implement Spivey's message queue server 120. Consequently, his combination ignores the very message flow that Spivey claims as inventive for reducing latency, as I further discuss below. Neither Petitioner nor Dr. Wicker explain how Spivey's message queue server device would operate in the absence of Spivey's other components or how a POSITA would achieve Spivey's reduction in latency without the same message flow.

264. Spivey discloses a number of embodiments, all of which seek to minimize latency by relying on the same "message flow"—that is, a process in which data is moved from "live data sources" to the message queue server 120, from the message queue server 120 to the live data access server, and from the live data access server to the user (i.e., "client") devices:



(EX-1017, Fig. 6, ¶¶65-69.)

265. Spivey explains that:

> Message flow 600 begins when live data is received from live data source(s) 140 at message queue server 120 in message 602….Message queue server 120, upon receiving message 602, may sort the data payload of message 602 into a topic queue….In some embodiments, message queue server 120, e.g. using topic decoder 320, may format the data payload or otherwise alter the structure of the data

> payload….Once the data payload is inserted into a topic queue, message queue server 120 may push the data payload to live data access server 130 in message 606. Again, this assumes that live data access server 130 has subscribed to the topic queue selected in message 604. Upon receipt of message 606, live data access server 130 may push the data payload to client device 110-*a* in message 608, assuming both that client device 110-*a* selected the topic corresponding to the topic queue of message 604 and that the connection between client device 110-*a* and the live data access server 130 is still open.

(EX-1017, ¶¶66-69.)

266. Spivey consistently describes this same message flow throughout all embodiments of its invention. (*See, e.g.*, EX-1017, Figs. 1, 7, 11, 12, 14, 17, 22, ¶¶36-38, 46-49, 75-79, 91-100, 108, 126-132, 160.)

267. The portion of Spivey's system that Petitioner proposes implementing in Herzog is one component of one embodiment <u>of the message queue server 120</u>. (EX-1017, ¶53.)

> Fig. 4 illustrates an embodiment of an operating environment 400. Operating environment 400 may illustrate a second embodiment ***of message queue server 120***. As shown in Fig. 4, ***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.

(EX-1017, ¶53 (emphasis added).) Dr. Wicker does not explain how this one component (message queue server device 410) would work without the other component

within Spivey's message queue server (live data server device 450). In addition, because this is merely one embodiment of the message queue server 120, Spivey still teaches that this alternative embodiment (and the components comprising that embodiment) should be applied using the same message flow and system illustrated and described throughout its specification in order to achieve a reduction in latency. (*See, e.g.*, EX-1017, Figs. 1, 6, 7, 11, 12, 17, 22, ¶¶36-38, 46-49, 75-79, 91-100, (all referencing or illustrating "message queue server 120"); *see also id.* Figs. 14, 17, 22, ¶¶108, 126-132, 160 (all referencing "message queue server 1420," which is described as an "embodiment[] of message queue server 120" at paragraph 107).)

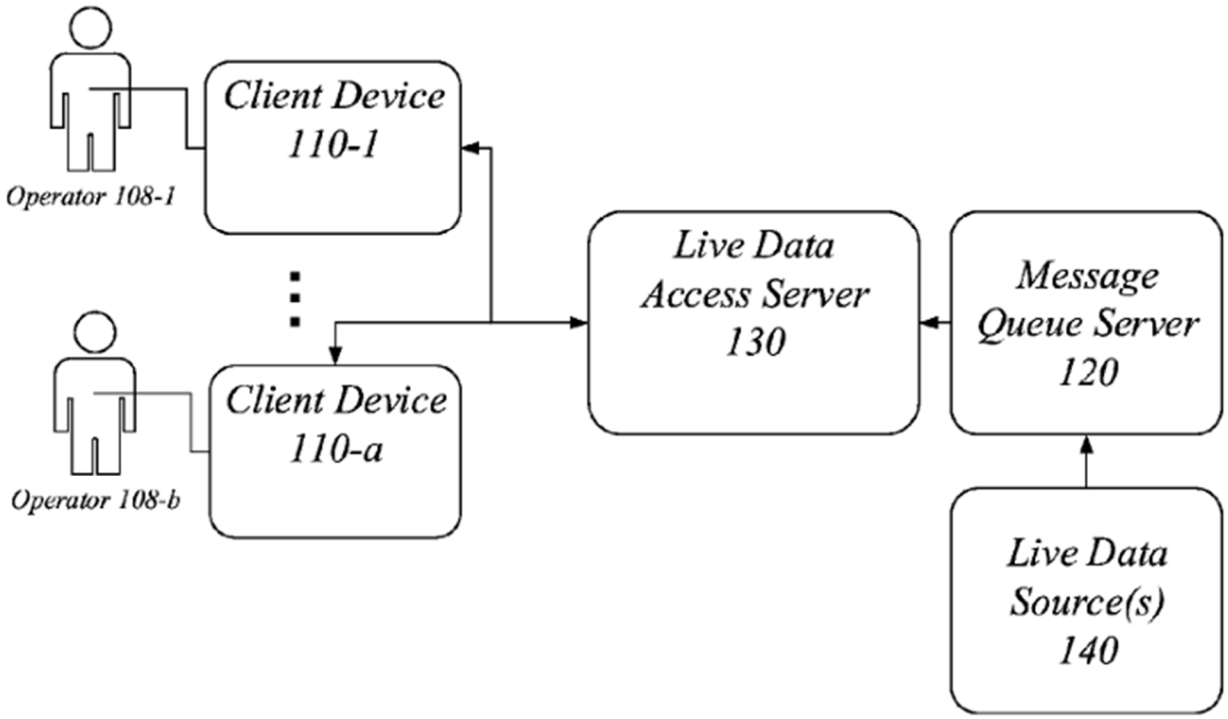
268. Dr. Wicker does not support his opinion that Spivey contemplates a different message flow that would provide the same benefits. On the other hand, Dr. Wicker seems to suggest that Spivey teaches a flow within which the message queue server might communicate directly with client devices, without a live data access server and without live data sources—which is not Spivey's teaching. (EX-1001, ¶144.) While failing to even address this modification of Spivey's message flow at this point in his declaration (*see* EX-1001, ¶144), Dr. Wicker includes minimal discussion elsewhere in his declaration, often repeating the same misquote of Spivey's specification. (*See* EX-1001, ¶99, ¶562, n.3; *see also id.*, ¶597, n.9, n.13.) Specifically, Dr. Wicker misquotes Spivey, stating that "Spivey also discloses that

the MQSD may be 'accessible to clients over the network directly.'" (EX-1001, ¶562, n.3; *see also id.*, ¶597, n.9, n.13.)

269. In fact, Spivey teaches that "message queue server 120 is ***not*** accessible to clients over the network directly." (EX-1017, ¶36.) Nowhere in the figures, specification, or claims, does Spivey describe an embodiment in which the message queue server (or message queue server device) delivers data to client devices without the use of a live data access server. And nowhere does Spivey suggest that such an altered flow would achieve the same reduction in latency.

270. In reviewing Dr. Wicker's minimal discussion of this issue, he cites the following disclosures from Spivey: ¶¶36, 38, 46, 53, 84, Fig. 4. But none of these disclosures teach removal of the live data server device or the live data access server. Neither do these disclosures disclose an embodiment lacking a live data access server.

271. Paragraphs 36, 38, and 46 all discuss "message queue server 120" or "message queue server 310," which is "a representative example of message queue server 120." (EX-1017, ¶¶ 36, 38, 46.) Message queue server 120 is illustrated in Figure 1 below, which shows the same message flow I describe above—i.e., from live data source to message queue server 120, from message queue server 120 to live data access server, and from live data access server to client devices:



(EX-1017, Fig. 1.) While Spivey explains that the message queue server 120 "may process live data to provide live data quickly to the client devices 110-1 through 110-a," this statement does not indicate that it is the server that provides data ***directly*** to the client devices. (EX-1017, ¶36.) It merely conveys what it states—that the server processes live data.

272. Dr. Wicker places much emphasis on the fact that Spivey states the system "*may*" include a live data access server 130. (EX-1001, n.3 (quoting EX-1017, ¶38).) By that same logic, Spivey's system "may" include a message queue server or not, despite the fact that the message queue server is critical and necessary to achieving Spivey's invention. (EX-1017, ¶¶35.) This language does not, in fact,

teach a POSITA how Spivey's system and alleged benefits would be achieved in the absence of a live data access server.

273. Dr. Wicker also relies on the ability of the message queue server to push data from topic queues to subscribers of the topics; however (and again), nothing in this stated ability teaches a POSITA that the subscribers are necessarily client devices and that the client devices are receiving the data directly from the message queue server. (EX-1017, ¶46.) A POSITA would not understand Spivey to implement the server in this way, when Spivey consistently teaches that the live data access server receives client subscription requests, subscribes to the message queue server topics itself, and conveys data from the message queues to the subscribed client devices. (*See, e.g.*, EX-1017, Figs. 1, 7, 11, 12, 14, 17, 22, ¶¶36-38, 46-49, 75-79, 91-100, 108, 126-132, 160.)

274. As I have explained above, the figure 4 embodiment and paragraph 53 merely describe an alternative embodiment of message queue server 120, in which it is implemented as two separate devices: a message queue server device 410 and a live data server device 450. (EX-1017, ¶53, Fig. 4.) Spivey explains that "[i]mplementing a message queue server as two separate devices may be useful to protect proprietary data that may be stored on live data server device 450 and/or to limit potentially harmful access to the live data server device." (EX-1017, ¶57.) But nothing in this alternative embodiment of the message queue server teaches a

POSITA to implement this embodiment outside of the message flow that Spivey consistently describes.

275. The final paragraph 84 that Dr. Wicker cites as support is no more illuminating. Spivey discloses a user interface that allows a client device to view and select topics available from a message queue server, to which the client wishes to subscribe. (EX-1017, ¶84.) Again, this is consistent with Spivey's repeated description of the live data access server subscribing on behalf of client devices to topics available at the message queue server. (*See, e.g.*, EX-1017, Figs. 1, 7, 11, 12, 14, 17, 22, ¶¶36-38, 46-49, 75-79, 91-100, 108, 126-132, 160.)

276. In what may be an attempted back-up argument that modifying Spivey's message flow would have been obvious, Dr. Wicker asserts in a footnote that a POSITA would have cut out the live data access server, because "reducing the amount of server transfers would reduce latency, which is Spivey's main purpose." (EX-1001, n.3.)

277. Dr. Wicker fails to acknowledge, however, the increase to latency when the live data access server is removed from Spivey's disclosed message flow. Spivey teaches that the live data access server provides a public interface between client devices and message queue server 120. (EX-1017, ¶38.)

> For example, live data access server 130 may allow operators 108 of client devices 110 to select topics related to a live event to subscribe to, and may retrieve the topic information and push the information to the

> client devices 110…. A client device, e.g. client device 110-1, may be usable by an operator, e.g. operator 108-1, to subscribe to and receive live data ***via live data access server 130***.

(EX-1017, ¶¶38-39 (emphasis added).) Spivey further teaches that by using the live data access server, data payloads can be immediately pushed to the subscribed live data access server, such that data payloads need no longer be stored on the message queue server. (EX-1017, ¶51.)

278. Assuming a POSITA were to reconfigure the message queue server device to even operate without the assistance of the live data access server (which Dr. Wicker does not explain), these operations which Spivey allocates to the live data access server would have to be picked up by the message queue server 120, thereby slowing down its operations and increasing latency. Rather than the message queue server's operations being focused on processing and pushing out data, the server would also need to manage all client subscriptions (as opposed to just subscriptions by the live data access server) and ensure that data payloads are conveyed from the correct message queues to the subscribed client devices. (*See* EX-1017, ¶¶66-69, 36-39.) This increased load on the message queue server would, in fact, increase latency and delay messages being received by the client devices.

279. Spivey repeatedly describes its reduction in latency (e.g., "at least two orders of magnitude shorter than conventional time periods") as a result of Spivey's message flow, including the live data access server. (*E.g.*, EX-1017, ¶96 (describing

the flow of the data payload from live data access server to the subscribed client devices); *see also id.*, ¶¶69-70, 100-101.)

280. That is, Petitioner and Dr. Wicker do not merely apply Spivey's disclosure to Herzog's system. Rather, they remove the message queue server device from the context of Spivey's invention and disclosed message flow, modify the message queue server device to include additional responsibilities and capabilities, and do so without regard for the impact of these modifications and without explaining why a POSITA would have made such modifications in the first place. The resulting system suffers from many problems.

b. Petitioner's proposed combination of teachings suffers from many problems.

281. Even assuming one would use Spivey's message queue server device while departing from all of Spivey's other teachings, the combination proposed by Petitioner and Dr. Wicker suffers from many problems.

282. ***First***, neither Petitioner nor Dr. Wicker have attempted to explain how the resulting system's message flow would operate or how a POSITA would have been able to predict its operation, since Spivey expressly teaches that the message queue server (one component within one embodiment being the message queue server device) will push topic-specific data payloads onto its queue to be conveyed ***via the subscribed live data access server*** to user/client devices. (EX-1017, Fig. 6, ¶¶66-69.) As I have stated above, this message flow is critical to Spivey's stated goal

to reduce latency and minimize delays. (Section VI(D)(1)(a), *supra*; EX-1017, ¶¶27-28, 95-96, Fig. 11.) In removing the message queue server device from this message flow, Petitioner and Dr. Wicker do not explain how Spivey's alleged reduction in latency would be achieved without the live data server device (which is also a part of Spivey's message queue server), or the subscribed live data access server. And a POSITA would not have been able to predict such a result or had a reasonable expectation of achieving such a result, when modifying Spivey's teachings.

283. Dr. Wicker misleadingly cherry picks quoted language from Spivey to suggest that Spivey teaches a direct connection by which the message queue server device "'provide[s] live data' to 'the subscribed connected client devices substantially simultaneously.'" (EX-1001, ¶591.) Spivey does not teach this. In fact, Spivey teaches:

> For example, ***live data access server 130*** may push the data payload to client devices 110 and to any other connected client devices that also requested that topic. In various embodiments, the data payload of the topic entry may be pushed to ***the subscribed connected client devices substantially simultaneously*** with the message queue server receiving the topic entry from a live data source….

(EX-1017, ¶96.) A POSITA would have understood that use of the live data access server was critical to Spivey's invention and achieving the benefits Spivey discloses, e.g., reduction in latency.

284. Dr. Wicker similarly fails to explain how Spivey's message queue server (or message queue server device) delivers data directly to client devices. As I discussed above, Spivey teaches that "message queue server 120 is not accessible to clients over the network directly." (EX-1017, ¶36.)

285. ***Second***, Petitioner and Dr. Wicker do not clearly articulate how Spivey's disclosure of a message queue server device would be combined with Herzog's teachings. At some points in the petition and Dr. Wicker's declaration, they first assert that a POSITA would have "implement[ed] Herzog's origin server ***as a websocket server***, as taught by Spivey," suggesting the origin server would be replaced with the message queue server or reconfigured to be a message queue server. (*See* Petition, 63; EX-1001, ¶143 (asserting a POSITA "would have looked to implement a system such as Spivey"), ¶601 (asserting the "MQSD can ***be*** an 'internet server'").) At other points of the petition and declaration, they instead assert that a POSITA would have "incorporat[ed]" Spivey's message queue server device "into Herzog's current data delivery system…at Herzog's origin server." (Petition, 63; EX-1001, ¶145.) Neither Petitioner's analysis nor Dr. Wicker's analysis clearly articulates precisely how Herzog's and Spivey's teachings would be implemented together—i.e., whether the resulting system would replace the origin server with Spivey's message queue server device or whether the message queue server device would be incorporated into the origin server in some unspecified way.

286. ***Third***, as I have discussed previously, Spivey does not disclose, contemplate, or relate to the delivery of live event video. (*See* Section VI(A)(1)(b), *supra*.) Implementing Spivey's message queue server device either at or in place of Herzog's origin server would render the resulting system inoperable, due to the unfettered provision of data by the message queue server device with no regard for the ongoing video/audio feed. Spivey does not teach how a message queue server device—once implemented in Herzog—would convey both a video/audio stream and a metadata stream to the content delivery network, as the origin server does. (EX-1015, ¶51.) Neither Petitioner nor Dr. Wicker have discussed how the references' teachings could be combined, such that the resulting system would still achieve Herzog's primary goal—synchronization of different latencies between a video/audio feed and metadata feed.

287. But implementing Spivey's message queue server device to immediately push out data payloads (even assuming the message flow issues above were resolved) would cause further issues in Herzog's system by speeding up only one of the feeds, causing even greater disparity between the video/audio feed and the data feed. Herzog's system still needs the projected latency period to perform synchronization between the video/audio data and metadata.

288. Dr. Wicker's suggestion that Herzog does not disclose back-end architecture to handle transmission of the video/audio data and metadata is incorrect.

(*See* EX-1001, ¶¶143, 147.) Herzog specifically discloses the origin server and describes its operations in detail. (EX-1015, ¶¶51, 69-74.) Thus, a POSITA would not have been at a loss as to the back-end architecture of Herzog or looking for hardware to implement its process. The alternative suggestion one could understand from Dr. Wicker's statement is that because Herzog does not disclose a socket server, a POSA would have looked to Spivey to implement a socket server:

> Herzog does not specifically disclose some backend server architecture supporting the data distribution—the claimed socket server. However, 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. Thus, it is my opinion that a POSA would have found it obvious to implement Herzog's origin server as a websocket server.

(EX-1001, ¶143.) But such reasoning—i.e., that a POSITA would have looked to Spivey for a socket server alternative to the HTTP polling Herzog already discloses, merely because Herzog does not disclose a socket server—appears to rely only on hindsight using the claim language.

289. Accordingly, for the reasons stated above, implementing a modified portion of Spivey's system within Herzog's system would not have been a simple substitution of known elements to obtain predictable results. (*See* EX-1001, ¶148.) The altered operation of Spivey—in which the message queue server device is implemented in a completely unique message flow lacking many of the necessary

components—was not known. And the proposed combination is neither clearly articulated (and, therefore, not a simple substitution), nor is it predictable, as it would decrease the functionality of Herzog's system, introduce new problems to be resolved, and render many of its intended benefits moot.

c. A POSITA would not have been motivated to combine the references' teachings.

290. In light of the many problems with Petitioner's proposed combination of teachings, which I discuss above, a POSITA would not have been motivated to modify and combine the references' teachings. (*See* Section VI(B)(1)(a)-(b), *supra*.) Dr. Wicker makes a number of assertions to the contrary, each of which mimics the same arguments made with respect to combining Ellis with Spivey and each of which I address below.

291. ***First***, Dr. Wicker's primary reason that a POSITA would have allegedly implemented Spivey's teachings within Herzog's system is to minimize latency. (EX-1001, ¶¶143-146.) Specifically, Dr. Wicker asserts a POSITA "looking to improve Herzog's latency issues, would have been looked to implement a system such as Spivey" in order to "achieve full-duplex communications between a client and a remote host with low latency." (EX-1001, ¶¶143-144.) There are many errors in this reasoning.

292. Herzog does not disclose any issues with latency or delay, and neither Petitioner nor Dr. Wicker explain what latency issues, in particular, Herzog's system

would have been understood to suffer from or the cause of such issues. Dr. Wicker broadens Spivey's actual teaching, alleging that "[a]s noted in Spivey, some live-event data delivery systems—such as Herzog—suffered from latency issues." (EX-1001, ¶144.) In fact, Spivey attributes the latency issues it sought to address as resulting from reliance on a database. (EX-1017, ¶28.) But Herzog does not teach storage of data to a database as a necessary step in its delivery system. Dr. Wicker refers to "Herzog's latency issues" without actually explaining why or how a POSITA would expect any implementation of Spivey's teachings to resolve those specific latency issues. (EX-1001, ¶144.) Indeed, as I have discussed above, they would only exacerbate the issues related to latency that Herzog sought to resolve. (Section VI(B)(1)(b), *supra*.) The application that resulted in the Herzog reference was filed in 2013, roughly a year before the application for Spivey was filed. (EX-1015, p.1; EX-1017, p.1.) Herzog's system is not one of the "legacy devices" that Spivey sought to improve upon, nor has Dr. Wicker shown it to be. (EX-1017, ¶96.)

293. Herzog is not directed to reducing latency; rather, Herzog particularly sought to provide a system that could synchronize the different latencies between a video/audio feed and metadata feed. (EX-1015, ¶¶3-4, 69-77, 82.) According to Herzog, in part because the video/audio feed enjoys a longer delay, the delayed time allows Herzog to retrieve metadata with a corresponding timestamp, such that it relates to the video/audio feed. (EX-1015, ¶¶58, 64-65, 71-77.) As the latency of the

video/audio feed ends, both the video/audio feed and the metadata are decoded and presented on the mobile device's display—i.e., where "[t]he metadata (the slave) is now synchronized to the video/audio data (the master)." (EX-1015, ¶¶58-59, 63.)

294. Furthermore, all of Spivey's disclosures of "full-duplex communications" that purportedly minimized delay, describe this type of connection as occurring between the live data access server 130 and the client devices. (EX-1017, ¶¶43, 61, 74, 92.) But Petitioner's proposed combination of teachings ignores the live data access server, instead incorporating only the message queue server device. That is, applying Spivey's limited, modified teaching in the way that Petitioner suggests does not include the "full-duplex connection" Spivey discloses.

295. For this reason, and the reasons I have stated above regarding altered message flow and the lack of a live data access server (Section VI(B)(1)(a)-(b)), neither Petitioner nor Dr. Wicker has shown how the proposed combination of teachings would actually minimize latency, and a POSITA would not expect it to.

296. Dr. Wicker additionally goes on to restate the arguments made in the petition, in nearly a word-for-word fashion. (*Compare* EX-1001, ¶¶143-148, *with* Petition, pp. 62-65.) Specifically, Dr. Wicker asserts that a POSITA would have been motivated to combine Herzog with Spivey, because Spivey provides a "complementary solution" describing the details of the backend architecture for transmitting event data to multiple users. (EX-1001, ¶147.) Dr. Wicker cites only the

abstract of Spivey and nothing else; he does not explain what he means by "complementary solution" or to what problem Spivey is allegedly a complementary solution. (*See id.*)

297. As I have discussed above, because Dr. Wicker modifies the teachings of Spivey by removing the message queue server device from the message flow Spivey discloses, a POSITA would not have been able to predict the resulting "solution" or how such a "solution" would have operated. (Sections VI(B)(1)(a)-(b), *supra*.)

298. Additionally, Dr. Wicker asserts that a POSITA would have been motivated to modify Herzog in view of Spivey, because both references related to "content delivery for real-time events." (EX-1001, ¶146.) However, as I have discussed previously, Spivey is not analogous to video delivery systems and contains no disclosure relevant to video delivery systems. (Section VI(A)(2), *supra*.) Furthermore, I am informed that the fact that references are drawn from the same broadly characterized field is insufficient evidence, alone, of a POSITA's motivation to combine.

299. Neither Petitioner nor Dr. Wicker have explained why or how a POSITA would have removed the message queue server device from the allegedly novel message flow Spivey touts or how any of the benefits taught by Spivey would have been preserved in the resulting system. Indeed, as I have discussed in detail, a

POSITA attempting to combine the teachings identified by Petitioner would render Herzog's system non-operational and lose the benefits of both Herzog's and Spivey's individual teachings. (Section VI(B)(1)(a)-(b), *supra*.)

300. Accordingly, a POSITA would not have been motivated to attempt such a combination of teachings.

2. A POSITA would not have been motivated to combine the teachings of Herzog and Ellis

301. A POSITA would not have been motivated to combine the teachings of Herzog and Ellis, nor would he have had a reasonable expectation of success in attempting such a combination.

302. Petitioner asserts that a POSITA would have been further motivated to modify Herzog in view of Ellis by "configure[ing] Herzog with Ellis's multiple event functionality," because users enjoy additional content options. (Petition, 65.) Dr. Wicker describes the proposed modification slightly differently, as "integrat[ing] Ellis's additional real-time information apparatus into Herzog/Spivey's system." (EX-1001, ¶¶149-153.) Both further assert that doing so "applies a known technique to a known device that was ready for improvement to yield a predictable result." (Petition, 66; EX-1001, ¶154.)

303. Petitioner's proposal ignores that Herzog discloses a system and method directed to the handling of a single event (e.g., an Indy race). (*See* Section V(C), *supra*.) Specifically, Herzog discloses "a system for delivering different types

of data to an end user, in which the data is obtained from different sources viewing ***a live event***, and synchronizing the data, so that the user can view the data in a comprehensive and orderly manner." (EX-1015, ¶9.) Herzog's method relies on time stamping video/audio data and metadata "with consistent (e.g., GPS) timing protocols" "at the video transmission point," despite the data being transmitted from different physical locations at the event. (EX-1015, ¶¶18, 53.) The "GPS based timing systems may be locked to the same satellite reference source." (EX-1015, ¶54.) Neither Petitioner nor Dr. Wicker explain how Herzog's pairing of the audio/video stream and metadata stream (which are associated with the same source) could accommodate or even be modified to apply to multiple events, particularly where such events would obviously yield wildly different GPS coordinates, timestamps, and other properties that would require adaptation and processing not taught by the references.

304. To the extent Dr. Wicker suggest integrating some unspecified "apparatus" disclosed by Ellis into Herzog's system (EX-1001, ¶153), he has further failed to explain how any part of Ellis's multi-tuner system could be modified to be made useful to a POSITA in Herzog's system. Ellis discloses a method and system for leveraging multiple tuners at a single user's television. (Section V(A), *supra*.) There is no indication or suggestion that such a system would be useful in Herzog's delivery of audio/video and metadata to users' mobile devices. And a POSITA

would not have seen Herzog's system, which is directed to mobile device streaming, as a "known device that was ready for improvement" by teachings relevant to a television tuning system.

305. Even if a POSITA were to implement multiple Herzog-like systems for multiple live events (which neither Petitioner nor Dr. Wicker sufficiently articulate), it is unclear how Herzog's method of matching GPS time stamps would be workable in this context, much less how these systems could be implemented in a coordinated manner. Accordingly, a POSITA would not have expected success and would not have been able to predict the results of attempting to combine such teachings in some unspecified manner.

3. **The asserted combination does not disclose or render obvious an "*event socket*" and/or *means for transmitting the first score information therethrough* or transmitting score information "*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*"**

306. Petitioner's Ground 2—just like Ground 1—relies on Spivey as allegedly disclosing an event socket at the message queue server device, which would then allegedly be implemented within Herzog's system. (EX-1001, ¶¶729-733.) Though, I note again that Petitioner and Dr. Wicker do not clearly articulate whether Spivey's disclosure of a message queue server device is purportedly being used to replace the origin server or is being incorporated into (i.e., added to) Herzog's origin server. (*Compare, e.g.*, Petition, 73 ("incorporated…into…"); EX-

1001, ¶725 ("incorporating…into") *with* Petition, 62-63 ("implement Herzog's origin server as a websocket server, as taught by Spivey" and "implement a system such as Spivey"); EX-1001, ¶¶126-127.)

307. Just as in Ground 1, Dr. Wicker identifies the message queue server device's topic queue as an event socket and alleges that Spivey teaches direct communication of data payloads from the topic queues to user/client devices via a "websocket connection." (EX-1001, ¶¶732-733.)

308. For the same reasons I discussed above in Section VI(A)(2), the combination of Spivey's and Herzog's teachings do not disclose or render obvious an "*event socket*" or transmitting information via a channel between an event socket and multiple viewer client devices in a synchronized manner. (*See* Section VI(A)(2), *supra*.)

309. Similar to Ground 1, Dr. Wicker does not provide any limitation-specific analysis of a POSITA's motivation to combine Herzog's and Spivey's teachings on this point, specifically, presumably relying on his more generalized discussion of motivation to combine. I have addressed the flaws in Petitioner's and Dr. Wicker's generalized motivation to combine the teachings of Herzog and Spivey, along with Ellis, above. (Section VI(B)(1)-(2), *supra*.) I have also further discussed the reasons a POSITA would not, in fact, have been motivated to combine

such teachings and why a POSITA would not have had a reasonable expectation of succeeding in combining such teachings. (Section VI(B)(1)-(2), *supra*.)

310. Accordingly, the proposed combination of prior art teachings does not disclose or render obvious this limitation.

VII. CLAIMS AND/OR LIMITATIONS NOT ADDRESSED IN THIS DECLARATION

311. It is my understanding that this declaration is being provided in support of Patent Owner's Preliminary Response. I have provided an analysis of key limitations not included in the prior art. The fact that I have not analyzed particular limitations or claims in no way indicates or supports that these limitations or claims are disclosed or rendered obvious by the prior art, and I reserve the right to supplement my analysis if asked to do so.

VIII. SECONDARY OBJECTIVE INDICIA OF NONOBVIOUSNESS

312. I understand that in determining obviousness of the claimed invention, objective evidence of secondary considerations pertaining to nonobviousness must be considered, including long-felt but unsolved needs, failure of others, commercial success, etc. I understand that such evidence is relevant where "nexus" exists—that is, where the evidence of secondary considerations is the direct result of the unique characteristics of the claimed invention. I understand that these indicia serve as independent evidence of nonobviousness, which can be highly persuasive in

avoiding the pitfalls of hindsight when evaluating what might otherwise appear to be a combination of known elements.

313. As demonstrated by my summary of the prior art, above, there was at least a long-felt and persistent need prior to the ’697 patent to improve upon prior art video streaming systems. (*See* Section III(C).) Prior to the claimed invention, conventional video streaming systems still suffered from significant viewer latency, *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. (EX-1006, 2:22-31.) The offset between different viewers can become especially significant when they watch videos of different qualities or use network connections having different bandwidths. (*See* EX-1006, 2:34-47.) I have further discussed conventional video streaming systems in Section III.C above.

314. Furthermore, conventional video streaming systems also had difficulties in providing relatively low latency copies of live video streams with relatively high quality (*e.g.*, high-definition HD and high bit rate, such as 2 to 5 megabits per second) or allowing for different classes or types of viewers to flexibly access copies of a live video stream via different live streaming formats, which may lead to different qualities of service. (EX-1006, 2:34-40.) For example, mainstream platforms such as Twitch.tv and Facebook had about 10-15 seconds of viewer latency. (EX-1006, 2:31-34.)

315. Thus, there was a long-felt need for an innovative system that enables broadcasters to create and deliver video and audio content without viewer latency issues, ensuring synchronization among varying numbers of viewers, and providing scalable and flexible access to live streams (*i.e.*, accommodating different classes or types of viewers and offering various qualities of service). (EX-1006, 2:31-64.)

316. Patent Owner's patented technologies helped transform the way video streaming is conducted, reducing, and often even eliminating the shortcomings that plagued conventional video streaming systems. More specifically, Patent Owner's patented technologies use a novel socket server architecture that allows actionable and contextual-matched information to be provided in a synchronized and efficient way with online streaming video. (*E.g.*, EX-1006, cl. 19.) For example, Patent Owner's patented technologies allow multiple users to connect to the same event socket, thereby achieving a one-to-many socket architecture. (*See* Section IV(A), *supra*.) Patent Owner's patented technologies' use of a one-to-many socket architecture was counter to the then-conventional understanding and use of sockets.

317. The patented invention further facilitated a variety of screen animations and other special effects graphics that may be generally associated with the video associated with a live stream, in a manner that is synchronized across multiple live streams with appreciably low latency. (*See* EX-1006, 5:56-63.) Accordingly, the '697 patented invention satisfied long-felt needs in the prior art.

318. I have additionally reviewed the evidence cited and summarized in Patent Owner's preliminary response, which further supports my opinion that the claimed invention of the '697 patent would not have been obvious. (*See* Patent Owner's Preliminary Response, Sections V.B.-V.C.)

319. Dr. Wicker states that he is unaware of any evidence of secondary considerations (EX-1001, Section XI), despite Petitioner's own praise of the claimed invention and subsequent copying of the claimed invention in Petitioner's own Genius Sports Platform. As shown below, communications between Petitioner and Patent Owner showed that since 2018, Petitioner had access to and had high praise for Patent Owner's claimed invention and was eager to promote it:

On Feb 24, 2018, 3:10 PM -0500, Ben Turner <ben.turner@geniussports.com>, wrote:

> Hey Kevin,
>
> Thanks for your time yesterday. It is great to see what you guys are doing with SportsCastr. Let's organise a follow up call for when I am back in Australia and I will have a think about some potential intro's that might be of interest.

(EX-2016)

On Mon, 7 May 2018 at 7:15 am, Ben Turner <ben.turner@geniussports.com> wrote:

> Hi Kevin, Andrew and Luke,
>
> I wanted to introduce you all to each other. As I travel around I find many great companies working in the sports digital/tech space and like to make introductions where synergies may exist. So...
>
> Kevin April (CEO) and Andrew Schupak (CFO) founded Sportscastr - https://sportscastr.com/ - they are based in NY and doing amazing work around sports broadcast with social commentators, graphics and real time chat.

(EX-2017)

320. Later, I understand that Patent Owner contends that the Petitioner launched its own sport's streaming platform and incorporated the claimed invention into its own products in *SportsCastr Inc. v. Genius Sports Ltd.*, No. 2:23-cv-04771-JRG (E.D. Tex.). (EX-2018.)

321. That Petitioner, itself, apparently recognized the novelty and benefits of the claimed invention further supports my opinion that the '697 patented invention was nonobvious.

IX. CONCLUSION

322. For at least the foregoing reasons, it is my opinion that Petitioner's grounds do not render obvious any of the challenged claims of the '697 patent.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true;

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

Executed this 10th day of December 2024 in Stuart, Florida.

Respectfully submitted,

Aviel Rubin, Ph.D.