US012389080B2

(12) **United States Patent**
April et al.

(10) **Patent No.: US 12,389,080 B2**
(45) **Date of Patent: *Aug. 12, 2025**

(54) **SYSTEMS, APPARATUS, AND METHODS FOR RENDERING DIGITAL CONTENT STREAMS OF EVENTS, AND SYNCHRONIZATION OF EVENT INFORMATION WITH RENDERED STREAMS, VIA MULTIPLE INTERNET CHANNELS**

(71) Applicant: **SportsCastr, Inc.**, Las Vegas, NV (US)

(72) Inventors: **Kevin April**, Lake Forest, IL (US); **Peter Azuolas**, Las Vegas, NV (US); **Philip Nicholas Schupak**, Brooklyn, NY (US); **Brian Silston**, New York, NY (US)

(73) Assignee: **SportsCastr, Inc.**, Las Vegas, NV (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **18/474,139**

(22) Filed: **Sep. 25, 2023**

(65) **Prior Publication Data**

US 2024/0022796 A1 Jan. 18, 2024

Related U.S. Application Data

(63) Continuation of application No. 17/027,219, filed on Sep. 21, 2020, now Pat. No. 11,770,591, which is a (Continued)

(51) **Int. Cl.**
H04N 21/61 (2011.01)
H04L 65/611 (2022.01)
(Continued)

(52) **U.S. Cl.**
CPC ***H04N 21/6125*** (2013.01); ***H04L 65/611*** (2022.05); ***H04L 65/65*** (2022.05);
(Continued)

(58) **Field of Classification Search**
CPC H04N 21/6125; H04N 21/2187; H04N 21/2343; H04N 21/4316; H04N 21/47217;
(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,251,294 A 10/1993 Abelow
5,999,908 A 12/1999 Abelow
(Continued)

FOREIGN PATENT DOCUMENTS

AU 2015222869 B2 7/2019
CA 2304286 A1 10/2001
(Continued)

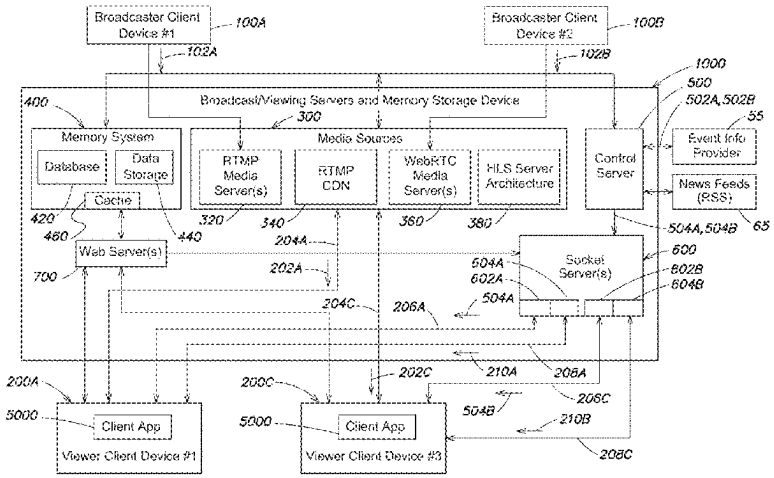
OTHER PUBLICATIONS

Sportradar Bookmaker SDK Deployment Guide, Jul. 3, 2014, 5 pages.
(Continued)

Primary Examiner — El Hadji M Sall
(74) *Attorney, Agent, or Firm* — Smith Baluch LLP

(57) **ABSTRACT**

Instructions are transmitted to first and second client devices to facilitate rendering of a video relating to an event (e.g., a broadcast of a live sporting event) together with event information (e.g., score information, online gaming information, team/player statistics). The first and second client devices are both instructed to receive the event information from an event socket of a socket server so that both client devices are synchronized and client-by-client latency is significantly reduced or mitigated. In one example, viewer latency between the event and the rendered video is on the order of two seconds or less. Event information is regularly updated at the event socket to also significantly reduce latency between the video and the event information. Event information may be rendered as one or more user-interactive
(Continued)



features overlaid on or adjacent to the video. When a user selects a user-interactive feature, a client device may obtain additional information relating to the event or redirect to a web site or an app.

30 Claims, 56 Drawing Sheets

Related U.S. Application Data

continuation of application No. 16/580,552, filed on Sep. 24, 2019, now Pat. No. 10,805,687, which is a continuation of application No. 16/267,887, filed on Feb. 5, 2019, now Pat. No. 10,425,697, which is a continuation of application No. PCT/US2017/045801, filed on Aug. 7, 2017.

(60) Provisional application No. 62/518,506, filed on Jun. 12, 2017, provisional application No. 62/485,878, filed on Apr. 14, 2017, provisional application No. 62/435,361, filed on Dec. 16, 2016, provisional application No. 62/371,558, filed on Aug. 5, 2016.

(51) **Int. Cl.**

H04L 65/65	(2022.01)
H04L 65/75	(2022.01)
H04L 65/752	(2022.01)
H04L 65/80	(2022.01)
H04N 21/2187	(2011.01)
H04N 21/2343	(2011.01)
H04N 21/431	(2011.01)
H04N 21/472	(2011.01)
H04N 21/4788	(2011.01)
H04N 21/845	(2011.01)

(52) **U.S. Cl.**

CPC ***H04L 65/75*** (2022.05); ***H04L 65/80*** (2013.01); ***H04N 21/2187*** (2013.01); ***H04N 21/2343*** (2013.01); ***H04N 21/4316*** (2013.01); ***H04N 21/47217*** (2013.01); ***H04N 21/4788*** (2013.01); ***H04N 21/8456*** (2013.01)

(58) **Field of Classification Search**

CPC H04N 21/4788; H04N 21/8456; H04L 65/611; H04L 65/65; H04L 65/75; H04L 65/80

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,434,621	B1	8/2002	Pezzillo et al.	
6,529,146	B1	3/2003	Kowalski et al.	
6,606,744	B1 *	8/2003	Mikurak	H04L 9/40 717/174
6,701,383	B1	3/2004	Wason et al.	
6,834,371	B1	12/2004	Jensen et al.	
6,839,059	B1	1/2005	Anderson et al.	
6,922,702	B1	7/2005	Jensen et al.	
7,133,834	B1	11/2006	Abelow	
7,228,349	B2	6/2007	Barone, Jr. et al.	
7,386,870	B2	6/2008	Lu	
7,458,093	B2	11/2008	Dukes et al.	
7,490,169	B1 *	2/2009	Ogdon	H04L 67/02 709/248
7,729,940	B2	6/2010	Harvey et al.	
7,882,533	B2	2/2011	Alao et al.	
8,000,993	B2	8/2011	Harvey et al.	
8,002,618	B1	8/2011	Lockton et al.	
8,010,986	B2	8/2011	Dakss et al.	
8,112,301	B2	2/2012	Harvey et al.	
8,128,503	B1	3/2012	Haot et al.	
8,149,530	B1	4/2012	Lockton et al.	
8,214,429	B2	7/2012	Chidel et al.	
8,341,662	B1	12/2012	Bassett et al.	
8,376,855	B2	2/2013	Lockton et al.	
8,549,574	B2	10/2013	Perlman et al.	
8,555,313	B2	10/2013	Newnam et al.	
8,595,186	B1	11/2013	Mandyam et al.	
8,656,042	B2	2/2014	Glasser et al.	
8,677,420	B2	3/2014	Cromarty et al.	
8,738,694	B2	5/2014	Huske et al.	
8,752,113	B1	6/2014	Good et al.	
8,775,501	B2	7/2014	Chidel et al.	
8,813,112	B1	8/2014	Cibula et al.	
8,813,125	B2	8/2014	Reisman	
8,850,053	B2	9/2014	Basso et al.	
8,858,313	B1	10/2014	Selfors	
8,870,639	B2	10/2014	Lockton et al.	
8,874,778	B2	10/2014	Xu et al.	
8,880,719	B2	11/2014	Cooper	
9,055,271	B2	6/2015	Verna et al.	
9,065,984	B2	6/2015	Arseneau et al.	
9,066,153	B2	6/2015	Pfeffer	
9,100,461	B2	8/2015	Lemmons et al.	
9,106,607	B1	8/2015	Lepeska et al.	
9,106,934	B2	8/2015	Horen et al.	
9,143,565	B2	9/2015	Hensgen et al.	
9,149,682	B2	10/2015	Dornbush et al.	
9,246,613	B2	1/2016	McKee et al.	
9,246,965	B1	1/2016	Stoica et al.	
9,251,852	B2	2/2016	Burns	
9,288,278	B2	3/2016	Panje et al.	
9,350,780	B2	5/2016	Good et al.	
9,357,246	B2	5/2016	Wright et al.	
9,392,314	B1 *	7/2016	Lewis	H04N 21/2408
9,393,485	B2	7/2016	Sullivan	
9,456,230	B1	9/2016	Bota et al.	
9,509,793	B2	11/2016	Brown et al.	
9,511,287	B2	12/2016	Lockton et al.	
9,516,390	B2	12/2016	Lau et al.	
9,584,858	B2	2/2017	Vinson et al.	
9,591,054	B2	3/2017	Thornburgh et al.	
9,641,566	B1	5/2017	Hiremath	
9,654,844	B2	5/2017	Kim et al.	
9,692,800	B2	6/2017	Gaunt et al.	
9,706,443	B2	7/2017	Oyman et al.	
9,716,918	B1	7/2017	Lockton et al.	
9,878,243	B2	1/2018	Lockton et al.	
9,905,082	B2	2/2018	Dengler et al.	
9,919,210	B2	3/2018	Lockton	
9,954,646	B1	4/2018	Marcin et al.	
9,973,785	B1	5/2018	Yang et al.	
10,015,224	B1	7/2018	Shen et al.	
10,105,596	B1	10/2018	Mann	
10,116,989	B1	10/2018	Shen et al.	
10,129,310	B1	11/2018	Brunning	
10,150,030	B2	12/2018	Perlman et al.	
10,226,698	B1	3/2019	Lockton et al.	
10,243,694	B1	3/2019	Marcin et al.	
10,313,412	B1	6/2019	Hall et al.	
10,326,814	B1	6/2019	Hall et al.	
10,327,040	B1	6/2019	Shen et al.	
10,397,291	B1	8/2019	Hall et al.	
10,425,697	B2	9/2019	April et al.	
10,437,551	B1	10/2019	Stanek et al.	
10,462,524	B2	10/2019	Barnett et al.	
10,484,730	B1	11/2019	Li et al.	
10,484,743	B2	11/2019	Cox	
10,535,177	B2	1/2020	Cornell	
10,554,324	B1	2/2020	Lim et al.	
10,601,914	B2	3/2020	Birrer et al.	
10,630,746	B1	4/2020	Shen et al.	
10,681,398	B1	6/2020	Marcin et al.	
10,715,860	B1	7/2020	Bartlett et al.	
10,721,543	B2	7/2020	Huske et al.	
10,735,783	B1	8/2020	Shen et al.	
10,740,305	B2	8/2020	Barthel et al.	
10,742,699	B1	8/2020	Shen et al.	

(56) **References Cited**

U.S. PATENT DOCUMENTS

10,805,687 B2 10/2020 April et al.
10,848,792 B2 11/2020 Evans
10,855,763 B2 12/2020 Birrer et al.
10,970,904 B1 4/2021 Kosmiskas
11,039,102 B1 6/2021 Marcin et al.
11,039,218 B1 6/2021 April et al.
11,048,764 B2 6/2021 Wexler et al.
11,051,049 B2 6/2021 Bustamante et al.
11,076,111 B1 7/2021 Ni et al.
11,076,188 B1 7/2021 Purushe
11,146,834 B1 10/2021 Shen et al.
11,153,581 B1 10/2021 Purushe
11,157,233 B1 10/2021 Stanek et al.
11,159,851 B2 10/2021 Stern et al.
11,178,447 B1 11/2021 Panter et al.
11,356,742 B2 6/2022 Azuolas et al.
11,405,658 B2 8/2022 Masterson
11,418,845 B2 8/2022 Wong et al.
11,425,178 B1 8/2022 Shen et al.
11,425,219 B1 8/2022 Lee et al.
11,445,246 B1 9/2022 Eatedali
11,451,883 B2 9/2022 Huske et al.
11,457,245 B1 9/2022 Bhatia et al.
11,470,361 B2 10/2022 Bustamante et al.
11,490,132 B2 11/2022 Bustamante et al.
11,533,543 B1 12/2022 Suh et al.
11,658,822 B1 5/2023 Engers et al.
11,770,591 B2 9/2023 April et al.
11,792,444 B2 10/2023 Bustamante et al.
11,871,088 B2 1/2024 Azuolas et al.
2001/0001160 A1 5/2001 Shoff et al.
2001/0039209 A1 11/2001 DeWeese et al.
2002/0145621 A1 10/2002 Nguyen
2004/0003101 A1 1/2004 Roth et al.
2004/0039788 A1 2/2004 Lim et al.
2004/0177002 A1 9/2004 Abelow
2006/0177200 A1 8/2006 Deutmeyer et al.
2006/0224761 A1 10/2006 Howarth et al.
2007/0033531 A1 2/2007 Marsh
2007/0043632 A1 2/2007 Abelow
2007/0204321 A1 8/2007 Shen et al.
2008/0022347 A1 1/2008 Cohen
2008/0052751 A1 2/2008 Cromarty et al.
2008/0062318 A1 3/2008 Ellis et al.
2008/0082922 A1 4/2008 Biniak et al.
2008/0168493 A1 7/2008 Allen et al.
2009/0049496 A1 2/2009 Morton
2009/0144785 A1 6/2009 Walker et al.
2009/0183215 A1 7/2009 McCartie et al.
2009/0222754 A1 9/2009 Phillips et al.
2010/0070345 A1 3/2010 Abelow
2010/0274848 A1 10/2010 Altmaier et al.
2010/0299703 A1 11/2010 Altman
2011/0083144 A1 4/2011 Bocharov et al.
2011/0086144 A1 4/2011 Arampongpun et al.
2011/0090960 A1 4/2011 Leontaris et al.
2011/0126236 A1* 5/2011 Arrasvuori H04N 21/4755 725/46
2011/0138429 A1* 6/2011 Schade H04N 7/17318 725/98
2011/0213655 A1 9/2011 Henkin et al.
2011/0247044 A1 10/2011 Jacoby
2011/0252156 A1 10/2011 Basso et al.
2011/0280540 A1 11/2011 Woodman
2012/0005224 A1 1/2012 Ahrens et al.
2012/0011267 A1 1/2012 Ma et al.
2012/0058808 A1 3/2012 Lockton
2012/0069131 A1 3/2012 Abelow
2012/0072845 A1 3/2012 John et al.
2012/0210348 A1 8/2012 Verna et al.
2012/0295686 A1 11/2012 Lockton
2013/0086278 A1 4/2013 Schmidt
2013/0194431 A1 8/2013 O'Connor et al.
2013/0198044 A1 8/2013 O'Connor et al.
2013/0198769 A1 8/2013 Bountour
2013/0218942 A1 8/2013 Willis et al.
2013/0222597 A1 8/2013 Brink et al.
2013/0225285 A1 8/2013 Lockton
2013/0227596 A1 8/2013 Pettis et al.
2013/0297706 A1 11/2013 Arme et al.
2013/0297721 A1 11/2013 Chen et al.
2013/0311595 A1 11/2013 Milatinovici et al.
2014/0040010 A1 2/2014 Garcia-Martinez
2014/0052540 A1 2/2014 Rajaram et al.
2014/0108534 A1 4/2014 Good
2014/0129680 A1 5/2014 Mukherjee
2014/0129942 A1* 5/2014 Rathod H04N 21/44226 715/720
2014/0161412 A1* 6/2014 Chase H04N 9/79 386/224
2014/0229992 A1 8/2014 Ellis et al.
2014/0245365 A1* 8/2014 Axelrod H04N 21/8126 725/109
2014/0281007 A1 9/2014 Lemmons et al.
2014/0282677 A1 9/2014 Mantell et al.
2014/0359075 A1 12/2014 Amidei et al.
2015/0067505 A1 3/2015 Metcalf et al.
2015/0127845 A1 5/2015 Phillips et al.
2015/0131845 A1 5/2015 Forouhar et al.
2015/0163379 A1 6/2015 Herzog
2015/0163562 A1 6/2015 Leventhal et al.
2015/0295726 A1* 10/2015 Bland H04L 67/02 709/203
2015/0341812 A1* 11/2015 Dion H04N 21/6373 370/252
2016/0006817 A1 1/2016 Mitic et al.
2016/0036910 A1 2/2016 Spivey
2016/0037215 A1 2/2016 Cardona
2016/0086108 A1 3/2016 Abelow
2016/0191493 A1* 6/2016 Bowman H04L 21/8126 725/109
2016/0249108 A1 8/2016 Sexton
2016/0255403 A1 9/2016 Gomes et al.
2016/0261927 A1 9/2016 Smolic et al.
2016/0271493 A1* 9/2016 Perlman H04L 65/752
2016/0294906 A1 10/2016 Levinson et al.
2016/0316262 A1 10/2016 Chen
2016/0360261 A1 12/2016 Makhlouf
2016/0366464 A1 12/2016 Rouady et al.
2016/0381109 A1* 12/2016 Barnett H04N 21/4622 709/231
2016/0381408 A1 12/2016 Triano et al.
2017/0026671 A1* 1/2017 Neumeier G06F 16/7847
2017/0034237 A1 2/2017 Silver
2017/0041686 A1 2/2017 Park et al.
2017/0099516 A1 4/2017 Barbulescu et al.
2017/0142200 A1 5/2017 Kodner et al.
2017/0185601 A1 6/2017 Qin et al.
2017/0185666 A1 6/2017 Pasternack et al.
2017/0188054 A1 6/2017 Ma et al.
2017/0250882 A1 8/2017 Kellicker
2017/0264961 A1 9/2017 Lockton
2018/0025586 A1 1/2018 Lockton
2018/0139260 A1 5/2018 Houle et al.
2018/0167427 A1 6/2018 Kedenburg, III
2018/0176621 A1 6/2018 Grubbs et al.
2018/0213266 A1 7/2018 Kwak et al.
2018/0307383 A1 10/2018 Faulkner et al.
2018/0367820 A1 12/2018 Abulikemu
2020/0021892 A1 1/2020 April et al.
2020/0098228 A1 3/2020 Amaitis et al.
2020/0111325 A1 4/2020 Lockton et al.
2020/0162796 A1 5/2020 Azuolas et al.
2020/0245017 A1 7/2020 Ganschow et al.
2021/0160584 A1 5/2021 April et al.
2021/0168462 A1 6/2021 April et al.
2021/0397847 A1 12/2021 Jayaram et al.
2022/0254379 A1 8/2022 Chang et al.
2022/0327830 A1 10/2022 Chang et al.
2022/0335720 A1 10/2022 Chang et al.
2023/0379531 A1 11/2023 Azuolas et al.
2024/0022796 A1 1/2024 April et al.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2024/0284014	A1	8/2024	Azuolas et al.
2024/0414412	A1	12/2024	Azuolas et al.

FOREIGN PATENT DOCUMENTS

EP	2127320	A2	12/2009
EP	2920708	A1	9/2015
EP	3063945	B1	9/2016
WO	1991012583	A1	8/1991
WO	2011149558	A3	3/2012
WO	2022170281	A1	8/2022

OTHER PUBLICATIONS

Sportradar Bookmaker SDK Developer Guide Sep. 7, 2015, accessed at https://sdk.sportradar.com/bookmaker/java/file?g=1aae9f9b-9e0b-40af-869b-380eede8806d on Feb. 17, 2023, 11 pages.

Sportradar brochure "Unified Odds Feed Integration Process" Oct. 15, 2021, accessed at https://iodocs.betradar.com/unifiedsdk/UOF_Integration_Process.pdf on Feb. 17, 2023, 14 pages.

Sportradar Drive Fan Behavior with Insights 2021 accessed at WaybackMachines https://sportradar.us/sports-media/insights/ on Apr. 14, 2021, 4 pages.

Sportradar Live Streaming of Sports Content with FASPStream to International Online Platforms, 2017, accessed at https://www.ibm.com/aspera/pdfs/Sportradar_AsperaCS_2017.pdf on Feb. 16, 2023, 2 pages.

Sportradar Live-Score Setup Document Apr. 2021, 8 pages.

Sportradar NFL V7 US API Portal, Feb. 7, 2023, 5 pages.

Sportradar Sports Data API Dec. 23, 2022, accessed at https://sportradar.com/sports-entertainment/content/sports-data-api/?lang=en-us on Feb. 16, 2023, 3 pages.

Sportradar Streams Live Data for Australian Open Broadcasts on Aspera, Digital Media World, accessed at WayBack Machines https://www.digitalmediaworld.tv/disrupt/2375-sportradar-streams-live-data-for-australian-open-broadcases-on-aspera on May 16, 2021, (2019) 3 pages.

Sportradar Supercharge Your OTT Solution with the Power of Data, accessed at Wayback Machine https://sportradar.us/sports-media/OTT on May 13, 2020, 3 pages.

Sportradar U.S. SEC Amendment No. 2 to Registration Statement, filed Sep. 7, 2021, Registration No. 333-258882 accessed at SEC Home » Search the Next-Generation EDGAR System» CompanySearch» on Feb. 16, 2023, 336 pages.

Sports—Personalized fan-first experiences that go beyond the field. Maestro 2020. Accessed at https://info.maestro.io/sports on Apr. 21, 2021. 3 pages.

Sports Betting. Sportradar. Accessed at https://sportradar.us/betting-services/ on Apr. 21, 2021. 4 pages.

Streaming, Grab Attention With Video, Genius Sports Website, 2022, accessed Apr. 13, 2023, 4 pages.

Styling Genius Live Player, GeniusSports Drop & Play Integration Guide, Nov. 28, 2022, 2 pages.

Understanding Match State, GeniusSports Wiki, Mar. 24, 2023, 2 pages.

Video Integration and Player User Guide Feb. 25, 2021, Genius Sports Wiki Website, 1 page.

Video Streaming API, GeniusSports Wiki Website, Mar. 24, 2021, 1 page.

Video Streaming Genius Sports Wiki Website, Mar. 26, 2021, 1 page.

Video-Streaming-v1, Swagger, Apr. 2019, 3 pages.

Video-V2, Swagger, Apr. 2019, 4 pages.

Virtuals, GeniusSports Drop & Play Integration Guide, Nov. 28, 2022, 1 page.

Wallstreet, John, Genius Sports-Caesars NFL Betting Live Stream Validates Vision, Dec. 21, 2022, Sportico, 4 pages.

Warehouse—Read Stream API Documentation—American Football!, GeniusSports Developer Centre Website, May 12, 2021, 14 pages.

Warehouse—Read Stream API Documentation—Basketball, GeniusSports Developer Centre Website, Jul. 30, 2017, 27 pages.

Washington, Betgenius launches live streaming service for sportsbooks. Genius Sports Oct. 8, 2019. Accessed at https://news.geniussports.com/betgenius-launches-live-streaming-service-for-sportsbooks/ on Apr. 30, 2021. 5 pages.

International Search Report and Written Opinion in International Patent Application No. PCT/US2017/045801 dated Oct. 30, 2019, 18 pages.

International Search Report and Written Opinion in International Patent Application No. PCT/US2018/033016 dated Aug. 6, 2018, 29 pages.

International Search Report and Written Opinion dated May 13, 2022 in International application No. PCT/US2022/014999 22 pages.

LiveStats Version 7—Data Streaming, Dec. 5, 2020 GeniusSports Support Centre, 3 pages.

Multibet Front-End Integration, GeniusSports Wiki, Feb. 13, 2023, 2 pages.

56 Bit Sportradar Case Study, Sportradar Time System 2022 accessed at https://www.56bit.com/case-studies/sportradar-time-system on Feb. 16, 2023, 5 pages.

Analyze & Optimize. Maestro 2020. Accessed at https://info.maestro.io/analyze-and-optimize on Apr. 21, 2021. 2 pages.

Hisayuki Ohmata et al., Hybridcast: A New Media Experience by Integration of broadcasting and Broadband, Itu Kaleidoscope (2013), 8 pages.

How Does Twitch Keep a Persistent Video Window over Several p. Stack Overflow, https://stackoverflow.com/questions/41286748/how-does-twitch-keep-a-persistent-video.window-over-several-p. 3 pages.

How Facebook Live Streams to 800,000 Simultaneous Viewers, High Scalability (Jun. 27, 2016), https:// highscalability.com/how-facebook-live-streams-to-800000-simultaneous.viewers/, 11 pages.

How to Create a WebSocket Server, WowzABLOG (Jun. 23, 2016), https://web.archive.org/web/20170926004104/https://www.wowza.com/docs/how-to-creat ··-a-websocket-server, 14 pages.

How to Stream to Facebook Live, TELESTREAM, https://www.telestream.net/telestream· support/wire-cast/how-to-stream-to-facebook-live.htm, 16 pages.

HTML5 Chat is Live!, Twitch Blog (Jun. 30, 2015), https://blog.twitch.tv/en/2015/06/30/html5-chat-is-live-345782edded0/, 4 pages.

Ian Sherr, Videogames Become a Spectator Sport, Wall St. J. (Nov. 24, 2013), http://online.wsj.com/article/SB10001424052702304607104579210140721840928.html, 4 pages.

Ida-Maria Tunturipuro, Building a Low-Cost Streaming System, May 14, 2015, Bachelor's Thesis, Helsinki Metropolia University of Applied Science, 84 pages.

Jake Smith, Facebook Rolls Out Live Video Streaming to All iPhone Users Across US, Zdnet (Jan. 28, 2016), https://web.archive.org/web/20160129091001/http://www.zdnet.com/article/facebook-rolls.out-live-streaming-to-all-iphone-users-across-US/, 1 page.

James Cook, Twitch Founder: We Turned A 'Terrible Idea' into a Billion-Dollar Company, Bus. INSIDER (Oct. 20, 2014), 15 pages.

James Trew, MLB At Bat 2012 App Update Brings New Stats Overlay, Ford SYNC integration and more, ENGADGET (Sept. 3, 2012), https://www.engadget.com/2012-09-03-mlb-at-bat-2012-app-update-brings-new-stats-overlay.html, 1 page.

Jan Ozer, Usstream, Justin.Iv, Livestream, and Bambuster: Streaming Unplugged, Streaming Media (Jun. 2012), web.archive.org/web/20131010063326/https:/www.streamingmedia.com/Articles/ReadArticle.aspx, 18 pages.

Jan Ozer, Ustream Review: Strong Customization, but a Weak Embedded Player, Streaming Media (Feb. 24, 2012), https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=80899, 7 pages.

Jason Cipriani, How to Start a Facebook Live Video, CNET (Jul. 8, 2016), https://www.cnet.com/tech/services-and-software/how-to-use-facebooks-live-video-feature/, 7 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Jason Maestas, Top Games for Jul. 2015 - Rocket League Edition, Twitch Blog (Aug. 10, 2015), https://blog. twitch.tv/en/2015/08/10/top-games-for-july-2015-rocket-league-edition. f278d248475a, 4 pages.
Jeff Baumgartner, Comcast's XI Blends Sports with Interactivity, Multichannel News (Jun. 19, 2015), https:// www.nexttv.com/news/comcast-s-xl-blends-sports-interactivity-391773, 10 pages.
Jie Deng et al., Behind the Game: Exploring the Twitch Streaming Platform, IEEE 14rn Int'l Work. Shop On Network and Sys. Support for Games (2015), 6 pages.
Jillian D'Onfro, Facebookjust proved how serious it is about live video with a huge product update, Business Insider (Apr. 6, 2016), https://www.businessinsider.com/facebook-live. video-update-2016-4, 12 pages.
John C. Tang et al., Meerkat and Periscope: I Stream, You Stream, Apps Stream for Live Streams, 4770 CHI (2016), 11 pages.
John Constine, Facebook Launches "Live" Streaming Video Feature, But Only for Celebrities, TECHCRUNCH (Aug. 5, 2015), https://techcrunch.com/2015/08/05/facescope/, 16 pages.
Jordan Golson, MLB At Bat Goes Live for 2012 with Improved Pricing Scheme, MACRUMORS (Feb. 29, 2012), https:// web.archive.org/web/20120302182806/http://www.macrumors.com:80/2012/02/29 /mlb-at-bat-goes-live-for-2012-with- improved-pricing-scheme, 6 pages.
Jordan Golson, MLB At Bat Updated for the 2013 Season, Local Game Video Still Blacked Out, MACRUMORS (Feb. 21, 2013), https://web.archive.org/web/20130225104539/http://www.macrumors.com:80/2013/02/21/ml b-at-bat- updated-dor-the-2013-season-local-game-video-still-blacked-out/, 9 pages.
Joshua Carroll, How to Live Stream a Sporting Event, Instructables (Nov. 22, 2015), http://www.instructables.com/id/How-to-live-stream-a-sporting-event-using-equipmen/, 6 pages.
Julia Greenberg, Zuckerberg Really Wants You to Stream Live Video on Facebook, WIRED (Apr. 6, 2016), https://www.wired.com/2016/04/facebook-really-really-wants-broadcas ·· watch-live-video/, 15 pages.
Julianna Reyes, How OneTwoSee's Baseball App is Getting in front of Millions of Comcast Subscribers, Technical. Ly (Jun. 30, 2015), https://technical.ly/startups/comast-mlb· baseball-onetwosee/, 2 pages.
Justin.tv's Live Video Broadcasting Architecture, High Scalability (Mar. 16, 2010), https://highscalability.com/justintvs-live-video-broadcasting-architecture/, 17 pages.
Karine Pires & Gwendal Simon, YouTube Live and Twitch: A Tour of User-Generated Live Streaming Systems, 225 MMSYS (2015), 6 pages.
Kate Abrosimova, What Technology Powers Periscope's Live Video Stream?, Medium (Jun. 5, 2016), https://web.archive.org/web/20170116004851/hhttps://medium.com/yalanti ·· mobile/what-technology-powers-periscopes-live-video-stream-1de663a13ebd, 8 pages.
Katie Collins, Periscope CEO: 'our app is just not very good at piracy,' Wired (Jul. 3, 2015), https://web.archive.org/web/20160807230614/http://www.wired.co.uk:80/article/periscope· ceo-kayvon-beykpour-wimbledon-game-of-thrones, 8 pages.
Kinji Matsumura et al., Personalization of Broadcast Programs using Synchronized Internet Content, ICCE (2010), pages.
Lexsan, Twitch and Youtube Settings Guide, XENFORO (Mar. 5, 2015), 3 pages.
Live Betting in 2009? Just a Warm up for 2010, Boolabus Blog (Jan. 2, 2010), https://web.archive.org/web/20140122054545/http://www.boolabus.com:80/2010/01/live, 4 pages.
Livestream Platform Tech Specs, https://web.archive.org/web/20160530045226/http://livestream.com:80/platform/tech, 6 pages.
Livestream, Livestream Broadcaster Pro User Manual (2015), 219 pages.
Livestream, New Livestream, 6 pages.
Livestream, Studio Live Production Switcher User Manual for Livestream Studio Version 1.1, 39 pages.
Livestream, Studio Surface User Manual (2015).
Lourdes Beloqui Yuste & Hugh Melvin, Client-side Multisource Media Streams Multiplexing/or HbbTV, 2NDINT'L Conf. on Consumerelecs (2012), 5 pages.
Lourdes Beloqui Yuste & Hugh Melvin, Enhanced IPTV Services Through Time Synchronisation, 14TH Int'l Symp On Consumer Elecs (ISCE) (2010), 10 pages.
Lourdes Beloqui Yuste et al., Effective Synchronisation of Hybrid Broadcast and Broadband TV, Int'l Conf. on Consumer Elecs. (2012), 2 pages.
M. Michalko et al., Building Video Streaming Platform for Educational Content Delivery, 331 ICETA (2014), 6 pages.
Mari Smith, Facebook Live: What Marketers Need to Know, Soc. MediaExam'r (Jan. 4, 2016), https://www.socialmediaexaminer.com/facebook-live-what-marketers-need-to-know/, 19 pages.
Mario Marques Da Silva, Multimedia Communications and Networking (2012), 494 pages.
Mark Newman, MLB.com At Bat launches into Spring, MLB (Feb. 25, 2014), https://www/mlb.com/news/mlbcom-at-bat-launches-into-spring/c-68295956, 4 pages.
Marketline, Betgenius Acquires Boolabus (2011), 1 page.
Matthew Panzarino, MLB At Bat for iOS brings Live Overlay of Gameday Data onto TV Broadcasts and More, Next Web (Aug. 31, 2012), https://thenextweb.com/news/mlb-at-bat-ios-updates-ford-syn ·· integration-social-sharing-highlight-videos, 3 pages.
Maury Brown, MLB Approves New Digital Media Company Spin-Off That Will Create Billions In New Revenues, FORBES (Aug. 13, 2015), https://www.forbes.com/ sites/maurybrown/2015/08/13/mlb-approves-new-digital-medi·· company-spin-off-that-will-create-billions-in-new-revenues/?sh=6d1ba6ce315d, 2 pages.
Miroslav Michalko et al., Protected Streaming of Video Content to Mobile Devices, JSAT 1 (2014), 7 pages.
Nick Wingfield, In E-Sports, Video Gamers Draw Real Crowds and Big Money, N.Y. Times (Aug. 30, 2014), https://www.nytimes.com/2014/08/31/technology/esports-explosionbring·· opportunity-riches-for-video-gamers.html, 8 pages.
Nick Wingfield, Julia Child Marathon to Stream on Twitch as Gaming Site Widens Focus, N.Y. Times (Mar. 15, 2016), https://www.nytimes.com/2016/03/16/technology/julia-chil·· marathon-tostream-on-twitch-in-bet-on-prerecorded-content. html, 4 pages.
Bbright, Periscope: First Look at Twitter's Live-Stream App, Mar. 26, 2015, https://www.youtube.com/watch? v=6mp4RkxvE0g, 1 page.
Brian MacSweeney, Jan. 5, 2015, Skybet Live Betting App, It's Half Time, https://www.youtube.com/watch? V=RYX4YspfYYM, 1 page.
Brian MacSweeney, Boolabus Betstream Snooker App, Apr. 22, 2011, https://www.youtube.com/watch?v=4Za2RhUC5sk, 1 page.
Brian MacSweeney, Boolabus Betstream, Betinplay football, Jan. 2, 2010, https://www.youtube.com/watch?v=1Shj1IR6W94, 1 page.
Brian MacSweeney, Skybet Live Betting App from Boolabus, Oct. 6, 2010, https://www.youtube.com/watch?v=E- LZ67zuPVY, 1 page.
Brian MacSweeney, Totesport Betstream live betting from Boolabus, Jun. 13, 2010, https://www.youtube.com/watch? V=UXJL9TNS_Hs, 1 page.
Brian MacSweeney, Totesport live betting marketing applications, Jun. 22, 2010, https://www.youtube.com/watch? V=Xs9sB-SOlul, 1 page.
Brian MacSweeney, Unibet Live Betting, Jan. 15, 2009, https://www.youtube.com/watch?v=kwnELhB2a30, 1 page.
Brian MacSweeney, www.Betstream.com - Sportingbet Live Betting Application, YouTube, Dec. 6, 2008, https:///www. youtube.com/watch?tv+E-LZ67zuPVY, 1 page.
Caleb Wojcik, Meerkat vs. Periscope Video Demo + How to Use Them, Mar. 31, 2015, https://www.youtube.com/ watch?v=PmkSad2udWI, 1 page.
Carrie Halperin, Times Minute, A (Very) Basic Twitch User's Guide, Aug. 26, 2014, https://www.youtube.com/watch?v=uVAu3PYasuY, 1 page.
Cbrit07, Livestream.com Tutorial, Dec. 7, 2009, https://www.youtube.com/watch?v=0zsYG7gXBVk, 1 page.
Chris Ducker, 4 Reasons Why Periscope is a HUGE Game Changer, Jul. 21, 2015, https://www.youtube.com/watch? v=IK_kmlYOWrc, 1 page.

(56) **References Cited**

OTHER PUBLICATIONS

CNN Business, First look at Periscope—the app that could kill Meerkat, Mar. 26, 2015, https://www.youtube.com/watch?v=wS6Lk_mf3eE, 1 page.

Cong Zhang & Jiangchuan Liu, On Crowdsourced Interactive Live Streaming: A Twitch. TVBased Measurement Study, 25TH ACM Workshop On Network and Operating Sys. Support for Digital Audio & VIDEO 55 (2015), arXiv preprint arXiv1502.04666v2, 2015, 7 pages.

Davelocity, New vs. Old Livestream.com, Apr. 19, 2012, https://www.youtube.com/watch?v=JTLpjD_0YX8, 1 page.

ExpatLive. TV, Honest Review of LiveStream HD550, Mevo Cameras and Livestream. com (Pt. 1), Feb. 15, 2017, https://www.youtube.com/watch?v=oCikHKLmDsg, 1 page.

Genius Sports Website—Integrate on Native Apps Using a Webview, 62 pages.

Genius Sports Website Virtuals, 1 page.

Halcyon Digital, Twitch Streaming Through OBS + How to get your stream key | Tutorial, Jul. 16, 2014, https://www.youtube.com/watch?v=KS3HNFV5dYs, 1 page.

Ireview4youri, Updated MLB At Bate 2011 iPhone App Review, iPhone App Review, Youtube Apr. 18, 2011; https://www.youtube.com/watch?v=6S0PM9iflp0, 1 page.

Periscope CEO on success and controversy of streaming app, https://www.youtube.com/watch?v=smXt58LrjRM, 2015, 1 page.

Philly NewTech, Jan. 2015: OneTwoSee—Chris Reynolds, Jason Angelides, Jan. 22, 2015, https://www.youtube.com/watch?v=054cf8_i7JU, 1 page.

Rich McCormick, Periscope Users Can Now Save Their Live-streamed Broadcasts Forever, TECHCRUNCH (May 5, 2016), https://www.theverge.com/2016/5/5/11595244/how-to-saveperiscopes-live-streams, 1 page.

Swagger—GL-Video-v3—https://exployer.api.geniussports.com/video/v3/PRODPRM/swagger.v3.1.0.json, 13 pages.

Swagger—Statistics API—/swagger/v2/swagger.json, 11 pages.

Twitch Engineering: An Introduction and Overview, Twitch Blog (Dec. 18, 2015), https://blog.twitch.tv/en/2015/12/18/twitch-engineering-an-introduction-and-overviewa23917b71a25/., 6 pages.

Twitter Dev, Twitter Flight 2015—Up Periscope by Kayon Beykpour and Sara Haider, Nov. 2, 2015, https://www.youtube.com/watch?v=xjC3ZKYG74g&list=PLFKjcMIU2Wsi599UeOXpODIJ60za0zr_h&index=3, 1 page.

Reasons Why Periscope Is Winning The Live-Streaming Battle, SETUP (Apr. 14, 2015), setup.US/blog/2015/4/14/5-reasons-why-periscope-is-winning-the-live-streaming-battle, 6 pages.

Adrien Danjou, How is the Sporting World Using Periscope Nowadays?, Snack Media (Apr. 2016), https://www. snack-media.com/2016/04/how-is-the-sporting-world-usin ·· periscope-nowadays/, 4 pages.

Apple Inc. http Live Streaming Overview (2016), 31 pages.

Apple Inc. Timed Metadata for http Live Streaming (2011), 8 pages.

Arjen Veenhuizen & Ray van Brandenburg, Frame accurate media synchronization of heterogeneous media sources in an HBB context, Media Synchronization Workshop (2012), 7 pages.

Betgenius Acquire Boolabus, BOOLABUSBLOG (Sept. 9, 2011), https://web.archive.org/web/20150504045439/http://www.boolabus.com:80/blog/, 10 pages.

Betstream Product Review in August Edition of "Egaming Review, "Boolabus Blog (Oct. 23, 2015), https://web.archive.org/web/20151023082029/http://www.boolabus.com:80/portfolio/, 4 pages.

BOOM More Transcode Servers., Twitch Blog (Jul. 8, 2015), https://blog.twitch.tv/en/2015/07/08/boom-more-transcode-servers-427bcbfb519/, 5 pages.

Brian Petrocelli, Attention Broadcasters: Channel Feed Beta is Now Available, Twitch Blog (Mar. 28, 2016), https://blog.twitch.tv/en/2016/03/28/attention-broadcasters-channe··feed-beta-is-now-available-28f7c3dff995/, 5 pages.

Brian Petrocelli, Channel Feed: Now with Comments!, Twitch Blog (Jul. 11, 2016), https://blog.twitch.tv/en/2016/07/11/channel-feed-now-with-comments-b7a0bc18e806/, 4 pages.

Building a Better Twitch- Updating to https, Twitch Blog (Mar. 9, 2016), https://blog.twitch.tv/en/2016/03/09/building-a-better-twitch-updating-to-https·cf06c7a02c3a/, 4 pages.

Carlos R. Morales et al., Development of a Multi-Platform High Performance Computing Streaming Video Distribution Cluster, AM. Soc'y For Eng'g Educ. (2013), 6 pages.

Cong Zhang & Jiangchuan Liu, On Crowdsourced Interactive Live Streaming: A Twitch. TV· Based Measurement Study, 25rn ACM Workshop On Network and Operating Sys. Support for Digital Audio & VIDEO 55 (2015), 6 pages.

David Carr, Amazon Bets on Content in Deal for Twitch, N.Y. Times (Aug. 31, 2014), ttps://www.nytimes.com/2014/09/01/business/media/amazons-bet-on-content-in-a-hub-for ·-gamers.html, 5 pages.

David Pierce, Twitter's Periscope App Lets You Livestream Your World, WIRED (Mar. 26, 2015), https://www.wired.com/2015/03/periscope/, 16 pages.

Douglas Soo, Twitch Engineering: An Introduction and Overview, Twitch Eng'g (Jan. 19, 2016), 11 pages.

Embed Scoreboard User Guide, Live Score App, https://web.archive.org/web/20150926085347/http://www.live-scopr·· app.com:80/userguide/embed-scoreboard, 6 pages.

Fidji Simo, Introducing New Ways to Create, Share and Discover Live Video on Facebook, Facebook Watch (Dec. 8, 2016), 13 pages.

From NHL GameCenter to Nhl. Tv- A Look at MLBAM's Relaunch of NHL Digital Properties, SPORTS Bus. J. (Feb. 26, 2016), https://www.sportsbusinessjoumal.com/Daily/Issues/2016/02/26/Technology/from-nhl· gamecenter-to-nhl-tv-a-look-at-mlbams-relaunch-of-nhl-digital-pr%E2%80%A6, 3 pages.

Genius Sports LTD's Answer to Complaint filed Jan. 26, 2024, in U.S. District Court, Eastern Division of Texas Marshall Division, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 54 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions filed Mar. 14, 2024, in U.S. District Court, Eastern Division of Texas Marshall Division, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 42 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-01, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 227 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-02, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 499 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-03, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 221 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-04, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 522 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-05, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 245 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-06, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 143 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-07, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 208 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-08, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 257 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-09, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 159 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-10, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 490 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-11, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 470 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-12, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 584 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-13, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 177 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit A-14, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 238 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-01, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 208 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-02, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 408 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-03, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 210 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-04, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 456 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-05, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 218 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-06, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 132 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-07, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 193 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-08, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 237 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-09, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 152 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-10, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)*v. *Genius Sports LTD*, 476 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-11, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 371 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-12, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 492 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-13, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 119 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit B-14, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 192 pages.

Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-01, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)*v. *Genius Sports LTD*, 477 pages.

Nick Wingfield, What's Twitch? Gamers Know, and Amazon Is Spending $1 Billion on It, N.Y. Times (Aug. 25, 2014), https://www.nytimes.com/2014/08/26/technology/amazo ·· nears-a-dealfor-twitch.html, 4 pages.

Noreen T.M., HTML5 Video Player Beta Begins Today, Twitch Blog (Jul. 14, 2016), https://blog.twitch.tv/ en/2016/07/14/html5-video-player-beta-begins-today-135dlb7baa65/, 5 pages.

Omar Bladonado & Paul Saab, Networking@Scale, May 2016-Recap, Eng'g Meta (May 24, 2016), https://engineering.fb.com/2016/05/24/networking-traffic/networking-scale-may- 2016-recap/?ref=highscalability.com, 13 pages.

OneTwoSee Provides Licensed, White-labeled Products that Help Some of the Biggest Names in Sports Engage and Monetize Their Audiences Through any Screen, ONETWOSEE (Jul. 6, 2016), https://web.archive.org/web/20160706140110/http://onetwosee.com:80/, 4 pages.

Periscope Versus Meerkat, Bus. INSIDER (Mar. 2015), https://web.archive.org/web/20150328235030/http://www.businessinsider.com:80/periscop ··-versus-meerkat-2015-03, 24 pages.

Peter Stewart, The Periscope Bible (2015), 61 pages.

Preston Smalley, Comcast Bolsters Sports Technology Team with Acquisition of OneTwoSee, COMCASTVOICES (Mar. 7, 2016), https://web.archive.org/web/20160415093127/http://corporate.com cast.com:80/comcast· voices/comcast-bolsters-sports-technology-team-with-acquisition-of-onetwosee, 4 pages.

Preston Smalley, Watch a Game Like Never Before with New Baseball Extras for the XI Platform, Comcast Voices (Jun. 29, 2015), https://web.archive.org/web/20150630175059/http://corporate.comcast.com:80/comcast·voices/watch-a-game-like-never-before-with-new-baseball-extras-for-the-x I-platform, 3 pages.

Rhys Hiltner, Go's March to Low-latency GC, Twitch Blog (Jul. 5, 2016), https://blog.twitch.tv/en/2016/07/05/gos-march-to-low-latency-gc-a6fa96f06eb7/, 13 pages.

Rich McCormick, Periscope Users Can Now Save Their Live-streamed Broadcasts Forever, TECHCRUNCH (May 5, 2016), https://www.theverge.com/2016/5/5/11595244/how-to-sav ·· periscopes-live-streams, 2 pages.

Richard Nieva, Facebook Live-streaming Brings the Love and the Wow, with a Side of Sepia, CNET (Apr. 6, 2016), https://www.cnet.com/tech/mobile/facebook-live-streaming-tool-adds·filters-and-reactions/, 6 pages.

Rip Empson, Baseball's Digital Trifecta and How America's Pastime is Setting the Pace for Sports Online, TECHCRUNCH (Nov. 4, 2013), https://web.archive.org/web/20131110013907/http://techcrunch.com:80/2013/11/04/baseballs-digital-trifecta-and-how-americas-pastime-is-setting-the-pace-for-sports-online/, 15 pages.

Shalini Kurian, High Performing Native Chat on Twitch, Twitch Blog (Jul. 15, 2016), https://blog.twitch.tv/en/2016/07/15/high-performing-native-chat-on-twitch-79c1492eca06/, 14 pages.

Skybet Go Live with the Betstream Advanced Football Module, Boolabus Blog (Oct. 12, 2015), https://web.archive.org/web/20151023033718/http://www.boolabus.com/2010/10/skybet-live-football/, 4 pages.

Sportingbet in:play, Boolabus Blog (Apr. 5, 2010), https://web.archive.org/web/20151023063447/http://www.boolabus.com/2010/04/sportingbet-inplay/, 3 pages.

Steve Kovach, Stream Live Baseball Games to Your iPad with MLB At Bat, Bus. INSIDER (Mar. 31, 2011), https://www.businessinsider.com/stream-live-baseball-games-with-mlb-at·bat-2011-3, 23 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Streaming Engine Documentation, WowzAMEDIA SYS., https://web.archive.org/web/20150311225020/http://www.wowza.com:SO/products/streaming-engine/documentation, 3 pages.
The Backbone of Periscope: Wowza Streaming Engine, WowzA BLOG (Nov. 17, 2015), https://web.archive.org/web/2017031416431/https://www.wowza.com/blog/the-backbone-of-periscope-wowza-streaming-engine, 4 pages.
The Beginner's Guide to Periscope for Bloggers, Fabulous Blogging, https://www.fabulousblogging.com/2015/08/the-beginners-guide-to-periscope-for-bloggers/, 18 pages.
Twitch API Overview, Twitch Dev (2017), https://web.archive.org/web/20170325170124/https://dev.twitch.tv/docs, pages.
Twitch Engineering: An Introduction and Overview, Twitch Blog (Dec. 18, 2015), https://blog.twitch.tv/ en/2015/12/18/twitch-engineering-an-introduction-and-overvie ·· a23917b71a25/, 9 pages.
Twitch iOS 9 Multitasking Support Is Here!, Twitch Blog (Oct. 14, 2015), https://blog.twitch.tv/en/2015/10/14/twitch-ios-9-multitasking-support-is-her·· a6364debed5 f/ 5 pages.
Twitch Live Annotations Faq, Twitch Blog (Jul. 1, 2014), https://blog.twitch.tv/en/2014/07/01/twitch-live-annotations-faq-6db54fadeede/, 6 pages.
Update: Chat Replay is Now Live!, Twitch Blog (Mar. 2, 2016), https://blog.twitch.tv/en/2016/03/02/update-chat-replay-is-now-live-aac0b82305b6/, 6 pages.
Watch and Chat in Full Screen with Theater Mode, Twitch Blog (Dec. 16, 2014), https://blog.twitch.tv/en/2014/12/16/watch-and-chat-in-full-screen-with-theater-mode-321238b7bd38/, 5 pages.
Wowza Media Systems, Tech-Guide: Building a Live Mobile Streaming App (2016), 19 pages.
Wowza Media Systems, the History of Streaming, 1 page.
Wowza Media Systems, Wowza Streaming Engine Overview (Feb. 2014), 26 pages.
Wowza Media Systems, Wowza Streaming Engine User's Guide 4.0 (2007-2014), 122 pages.
Wowza Media Systems, Wowza Streaming Engine User's Guide 4.7.5 (2007-2018), 97 pages.
Wowza Media Systems, Wowza Streaming Engine User's Guide 4.I (2006-2015), 120 pages.
Yoree Koh & Evelyn M. Rusli, Twitter Acquires Live-Video Streaming Startup Periscope, Wall St. J. (Mar. 9, 2015), http://www.wsj.com/articles/twitter-acquires-live-vide·· streaming-startup-periscope-1425938498, 2 pages.
Genius Sports launches In-Play Multi Bet for the next generation of same game parlay bets Genius Sports Press, Oct. 5, 2022, 3 pages.
Genius Sports Live Sports Data Integration, GeniusSports Wiki, Jul. 5, 2022, 2 pages.
Genius Sports Live Sports Data Pull APIs, GeniusSports Wiki, Jul. 5, 2022, 2 pages.
Genius Sports Live Sports Data Push API via ably, GeniusSports Wiki, Nov. 28, 2022, 7 pages.
Genius Sports slashes cost and lowers latencies for last-mile live sports data delivery, Ably, Aug. 9, 2022, 8 pages.
Genussports, Genius Sports Agrees to Official Data, Trading & LiveStreaming Partnership with Bally's Interactive, Aug. 25, 2022, 3 pages.
GL-Video-v2, GeniusSports Explorer API Website; accessed Mar. 10, 2023, 28 pages.
GL-Video-v3, Swagger, Feb. 2023, 9 pages.
Ha, Major League Baseball's "At Bat" App Gets updated to Supported Expanded Instant Replay. TechCrunch Mar. 31, 2014. Accessed at https://techcrunch.com/2014/03/31/mlb-at-bat-expanded-instant-replay/. 4 pages.
Halpin Introducing Extensions: A Streaming Revolution, Twitch, Aug. 31, 2017, 5 pages, obtain at website: https://blog.twitch.tv/en/2017/08/31/introducing-extensions-a-streaming-revolution-c31762addcd9/ on Jun. 7, 2023, 5 pages.
Hashemizadehnaeini, Transcoding H.264 Video via FFMPEG encoder. Corso di Laurea Magistrale in Ingegneria delle Telecomunicazioni Politecnico Di Milano. Thesis, 2015, p. 1-100.
How Genius Sports delivers realtime data to their customers' frontend at speed and at scale, Webinar, May 2021, Ably, 3 pages.
How to Integrate It, GeniusSports Drop & Play Integration Guide, Nov. 28, 2022, 5 pages.
In-Play MultiBet Unlock new in-play revenues, GeniusSports Website, Mar. 13, 2023, 4 pages.
Integration Guide—Genius Live Player, GeniusSports Drop & Play Integration Guide, Nov. 28, 2022, 1 page.
Integration Schema—Integration Process GeniusSports Wiki, May 29, 2023, 4 pages.
Integration Schema—Integration Service—Interface, GeniusSports Wiki, Dec. 16, 2020, 2 pages.
Integration Schema—Integration Service GeniusSports Wiki, Apr. 13, 2022, 16 pages.
Integration Schema—Service Reliability and Availability, GeniusSports, Mar. 29, 2023, 2 pages.
Integration Schema GeniusSports Wiki, Oct. 11, 2019, 2 pages.
International Search Report and Written Opinion in International Patent Application No. PCT/US2017/045801 mailed Oct. 30, 2019, 18 pages. Opinion in International Patent Application No. PCT/US2018/033016 mailed SGIC-002WO01), 29 pages.
International Search Report and Written Opinion in International Patent Application No.: PCT/US2018/033016 mailed Aug. 6, 2018, 29 pages.
International Search Report and Written Opinion mailed May 13, 2022 in International application No. PCT/US2022/014999, 22 pages.
Introducing Drop and Play (Medium container), BetGenius Website, Jul. 5, 2022, 3 pages.
LA Techwatch, LA Startup Maestro Just Raised $3M to Broadcast eSports at Scale. LA TechWatch Sep. 19, 2017. Accessed at https://www.latechwatch.com/2017/09/la-startup-maestro-just-raised-3m-broadcast-esports-scale/. 13 pages.
Lemire "Sportradar Launches Real-Time Data Product for Media Companies" Sports Business Journal, Oct. 27, 2020, 1 page.
Live Sports Data Integration—API Explorer for Genius Ably Push Feeds, GeniusSports Wiki, Mar. 20, 2023, 5 pages.
Live Sports Data Integration—Authenticating against Genius Sports APIs, GeniusSports Wiki, Aug. 12, 2022, 2 pages.
Live Sports Data Integration—Match State Platform Access Control API, GeniusSports Wiki, Aug. 12, 2022, 3 pages.
Live Sports Data Integration—Statistics API, GeniusSports Wiki, Aug. 18, 2022, 5 pages.
Version 7—Data Streaming, Dec. 5, 2020 GeniusSports Support Centre, 3 pages.
Livestream—READ—FAQs, GeniusSports Developer Center Website, Jul. 31, 2017, 4 pages.
Menmuir Betgenius Launces 'Next Level' Sportbook Streaming Service, SBCNews, Oct. 8, 2019, 2 pages.
MLB App. MLB 2021. Accessed at https://www.mlb.com/apps/mlb-app on Apr. 21, 2021. 1 page.
Multibet Back-End Integration—Market Lifetime, GeniusSports Wiki, Feb. 13, 2023, 2 pages.
Multibet Back-End Integration—Multibet API, GeniusSports Wiki, Mar. 15, 2023, 10 pages.
Multibet Back-End Integration, GeniusSports Wiki, Feb. 13, 2023, 4 pages.
Multibet Back-End Integration, Multibet Sportsbook Integration (V3), GeniusSports Wiki, Feb. 13, 2023, 5 pages.
Multibet Front-End Integration—Authenticating User-SessionUsing a Proxy Service, GeniusSports Wiki, Feb. 13, 2023, 2 pages.
Multibet Front-End Integration—UI Widget Customization, GeniusSports Wiki, Feb. 13, 2023, 4 pages.
Multibet Front-End Integration, GeniusSports Wiki, Feburary 13, 2023, 2 pages.
Perez, Live Video Viewing up 86% over last year in MLB's At Bat app, thanks for Addition of multitasking. TechCrunch Apr. 15, 2016. Accessed at https://techcrunch.com/2016/04/15/live-video-viewing-up-86-over-last-year-in-mlbs-at-bat-app-thanks-to-addition-of-multitasking/. 4 pages.
Perez, MLB.com At Bat and NHL are first to Launch Personalized App Icons on IOS 10.3. TechCrunch Mar. 28, 2017. Accessed at

(56) **References Cited**

OTHER PUBLICATIONS

https://techcrunch.com/2017/03/28/mlb-com-at-bat-and-nhl-are-first-to-launch-personalized-app-icons-on-ios-10-3/. 5 pages.
Read-Read a stream of live game events.GeniusSports Developer Centre Website, Nov. 2, 2016, 28 pages.
References, GeniusSports Drop & Play Integration Guide, Nov. 28, 2022, 1 page.
Risk Services Integration, GeniusSports Wiki , Mar. 20, 2023, 10 pages.
SmartStream Integrations, GeniusSports Wiki, Oct. 26, 2022, 4 pages.
Sportradar—Data Services 2019, accessed at waybackmachine https://www.sportradar.com/rights-holder-solutions/data-services/ on Jun. 4, 2019, 3 pages.
Sportradar "Sportradar Accelerates Data and Video on AWS, Launches Simulated Reality Innovation in 11 Days" Dec. 2, 2020, accessed at https://aws.amazon.com/solutions/case-studies/sportradar-case-study/ on Dec. 20, 2020, 5 pages.
Sportradar Aspera, Live Streaming of Sports Content with FASPStream to International Online Platforms, 2017, assessed at https://www.IBM.com/aspera/pdfs/Sportradar_AsperaCS_2017.pdf on Feb. 24, 2023, 2 pages.
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-18, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 147 pages (SRAD-EX-C-18-Abulikemu).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-20, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 161 pages (SRAD-EX-C-20-Kaazing).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-21, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 294 pages (SRAD-EX-C-21-WebRTC).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-22, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 276 pages (SRAD-Ex-C-22-RTP).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-23, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 299 pages (SRAD-EX-C-23-WebSocket).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-24, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 166 pages (SRAD-EX-C-24-AppleHLS).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-25, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 171 pages (SRAD-EX-C-25-Betstream).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-26, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 65 pages (SRAD-EX-C-26-FacebookLive).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-27, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 76 pages (SRAD-EX-C-27-MLBAtBat).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-28, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 66 pages (SRAD-EX-C-28-OneTwoSee).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-01, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 100 pages (SRAD-EX-D-01-Lu).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-02, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 257 pages (SRAD-EX-D-02-Dukes).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-03, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 182 pages (SRAD-EX-D-03-Mandyam).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-04, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 136 pages (SRAD-EX-D-04-Cromarty).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-05, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 115 pages (SRAD-EX-D-05-Cooper).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-06, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 139 pages (SRAD-EX-D-06-Basso).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-07, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 116 pages (SRAD-EX-D-07-Sullivan).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-08, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 121 pages (SRAD-EX-D-08-Bota).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-09, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 80 pages (SRAD-EX-D-09-Dengler).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-10, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 117 pages (SRAD-EX-D-10-Ma).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-11, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 68 pages (SRAD-EX-D-11-Mukherjee).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-12, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 74 pages (SRAD-EX-D-12-Ellis).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-13, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 170 pages (SRAD-EX-D-13-Herzog).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-14, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 290 pages (SRAD-EX-D-14-Mitic).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-15, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 163 pages (SRAD-EX-D-15-Spivey).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-16, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 480 pages (SRAD-EX-D-16-Levinson).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-17, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 207 pages (SRAD-EX- D-17-Park).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-18, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 149 pages (SRAD-EX-D-18-Abulikemu).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-20, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 169 pages (SRAD-EX-D-20-Kaazing).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-21, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 406 pages (SRAD-EX-D-21-WebRTC).

(56) **References Cited**

OTHER PUBLICATIONS

Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-22, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 244 pages (SRAD-Ex-D-22-RTP).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-23, May 6, 2024, Case No. 2:23-cv-00472-. Jrg, Sportscastr Inc. (dba Panda Interactive) v. Sportradar Group AG and Sportradar AG, 283 pages (SRAD-EX- D-23-WebSocket).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-24, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 220 pages (SRAD-EX-D-24-AppleHLS).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-25, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 385 pages (SRAD-EX-D-25-Betstream).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-26, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 69 pages (SRAD-EX-D-26-FacebookLive).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-27, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 81 pages (SRAD-EX-D-27-MLBAtBat).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit D-28, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 69 pages (SRAD-EX-D-28-OneTwoSee).
Sportradar Group AG and Sportradar AG's p.r. 3-3 AND 3-4 Invalidity Contentions, in Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, filed May 6, 2024, 66 pages.
Twitch Blog (Aug. 10, 2015), https://blog.twitch.tv/en/2015/08/10/top-games-for-july-2015-rocket-league-editionf278d248476a/, 4 pages (Twitch_Blog_Aug_10_2015).
Vanessa Wang et al., The Definitive Guide To HTML5 Websocket (2013), 200 pages (Wang_SRAD_00009135).
Webrtc, Tutorials Point (2015), 34 pages (WebRTC_tutorials_point_2015).
Andrew S. Tanenbaum & David J. Wetherall, Computer Networks (5th Ed. 2011), 962 pages (Tanenbaum_Wetherall_2011).
Andy Harris, HTML5 and CSS3 All-In-One for Dummies (3d Ed. 2014), 1107 pages (Harris_HTML5_2014).
Betgenius, 188bet Proposal: Advanced Football Scorecentre (Jun. 12, 2012) ("Advanced Football Scorecentre"), 32 pages (Betgenius_188bet_2012_SRAD_00000812).
Betradar Product Brochure 2011/2012 (2011), 36 pages (Betradar_Product_Brochure_2011).
Betradar, Betradar—Feature Update: Snooker now available on Live Scout service (Dec. 2013) 12 pages (BETRADAR_Feature_update_Snooker_Live_Scout_2013).
Betradar, Betrader Help (Oct. 2012) 46 pages (Betradar_Help_Oct_2012).
Betradar, Bowls Scout Client Specification, 7 pages (BETRADAR_Bowls_Scout_SRAD_00009616).
Betradar, Snooker Scout Client (2013) 9 pages (BETRADAR_Snooker_Scout_Client_2013).
Betrader, Betradar—Feature Update: Bowls now available on Live Odds & Live Scouting services Oct. 2014, 12 pages (Betradar_Feature_Update_2014).
Boolabus Blog (May 4, 2015), https://web.archive.org/web/20150504045439/http://www.boolabus.com:80/blog/, ("Boolabus Blog"), 10 pages (Boolabus-Blog_05-04-2015_SRAD_00000958).
Brian MacSweeney & Peter Hall, Betfred Inplay Application Development: Technical Specification (Jun. 2, 2009) ("Inplay Technical Specification"), 28 pages (MacSweeney_Hall_SRAD_00000784).
Genius Sports LTD's Preliminary Claim Constructions and Extrinsic Evidence Pursuant to P.R. 4-2 filed Aug. 2, 2024, in U.S. District Court, Eastern Division of Texas Marshall Division, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 19 pages (Genius_Proposed_Claim_Constructions_08-22-2024_npl).
https://www.telestream.net/telestream-support/wire-cast/how-to-stream-to-facebook-live.htm, 16 pages (TELESTREAM_How_to_Stream_to_Facebook_live).
International Search Report and Written Opinion in International Patent Application No .: PCT/US2022/015713 mailed Jun. 2, 2022), 14 pages.
International Search Report and Written Opinion in PCT/US2024/028064 mailed Jun. 24, 2024, 16 pages.
Kaazing WebSocket Gateway: Detailed Technical Overview (Jun. 2010) 34 pages (Kaazing_Websocket_Gateway_June_2010).
Matthew Prince, CloudFlare Now Supports WebSockets, Cloudflare Blog (Aug. 5, 2014), https://blog. cloudflare.com/cloudflare-now-supports-websockets/, 7 pages (Prince_CloudFlare_SRAD_00009053).
MLB At Bat for iOS brings Live Overlay of Gameday Data onto TV Broadcasts and More, Next Web (Aug. 31, 2012), https://thenextweb.com/news/mlb-at-bat-ios-updates-ford-sync-integration-social-sharing-highlight-videos, 3 pages (MLB_at_Bat_for_IOS_Aug_31_2012).
Real-Time Second Screen Connectivity, Kaazing, https://web.archive.org/web/20130815224633/https://www.kaazing.com/solutions/real-time-second-screen-connectivity/, 2 pages (Real-time_Second_Screen_Kaazing).
Richard Clark, Whitepaper: Real-Time Data, KAAZING (2014) 10 pages (Clark_Whitepaper_Real-time_Data_2014).
Rob Manson, Getting Started With Webrtc (2013), 114 pages (Manson_Getting_Started_w_WebRTC-2013).
Salvatore Loreto & Simon Pietro Romano, Real-Time Communication With Webrtc (2014), 163 pages (Loreto_Romano_Real-time_Comm_w_WebRTC_2014).
Schulzrinne et al., RTP: A Transport Protocol for Real-Time Applications (2003), 89 pages (Schulzrinne_RTP_Transport_2003).
Second Screen App, KAAZING, https://web.archive.org/web/20130818005342/http://developer.kaazing.com/portfolio/second-screen-application/, 2 pages (Second_Screen_App_5342).
SEcond Screen App. KAAZING, https://web.archive.org/web/20130914172647/http://developer.kaazing.com:80/portfolio/second-screen-application, 2 pages (SEcond_Screen_App_2647).
Socket. IO Broadcasting events, SOCKET.IO, https://socket.io/docs/v4/broadcasting-events/, 3 pages (Socket_IO_SRAD_00008933).
Socket.IO Docs, SOCKET.IO https://web.archive.org/web/20150401041027/http:/socket.io/docs, 2 pages (Socket_IO_SRAD_00009039).
Socket.IO Documentation, SOCKET.IO, https://socket.io/docs/v4/, 4 pages (Socket_IO_SRAD_00009060).
Socket.IO on PubNub, PUBNUB https://web.archive.org/web/20150924221323mp_/http:/www.pubnub.com/socket.io, 3 pages, (Socket_IO_PubNub_SRAD_00009064).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-01, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 158 pages (SRAD-EX-A-01-Lu).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-02, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 302 pages (SRAD-EX-A-02-Dukes).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-03, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 144 pages (SRAD-EX-A-03-Mandyam).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-04, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 166 pages (SRAD-EX-A-04-Cromarty).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-05, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba v. Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 171 pages (SRAD-EX-A-05-Cooper).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-06, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr*

(56) **References Cited**

OTHER PUBLICATIONS

Inc. (dba Panda Interactive) v. *Sportradar Group AG and Sportradar AG*, 190 pages (SRAD-EX-A-06-Basso).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-07, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 166 pages (SRAD-EX-A-07-Sullivan).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-08, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 93 pages (SRAD-EX-A-08-Bota).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-09, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 56 pages (SRAD-EX-A-09-Dengler).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-10, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 73 pages (SRAD-EX-A-10-Ma).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-11, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 56 pages (SRAD-EX-A-11-Mukherjee).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-12, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 133 pages (SRAD-EX-A-12-Ellis).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-13, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 223 pages (SRAD-EX-A-13-Herzog).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-14, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 222 pages (SRAD-EX-A-14-Mitic).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-15, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 107 pages (SRAD-EX-A-15-Spivey).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-16, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 692 pages (SRAD-EX-A-16-Levinson).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-17, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 284 pages (SRAD-EX-A-17-Park).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-18, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 193 pages (SRAD-EX-A-18-Abulikemu).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-20, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 158 pages (SRAD-EX-A-20-Kaazing).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-21, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 244 pages (SRAD-EX-A-21-WebRTC).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-22, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 288 pages (SRAD-Ex-A-22-RTP).
A Brief History of the Internet, Online Lib. Learning Center. (Feb. 9, 2024), https://www.usg.edu/galileo/skills/ unit07/internet07_02.phtml ("Online Library"), 2 pages (A_Brief_History_ONLINE_LIB_2024).
Andy Salo, MPEG DASH: A Technical Deep Dive and Look at What's Next, RGB Networks, (2012) ("Salo"), 13 pages (Andy_Salo_MPEG_RGB_2012).
Brijesh Kumar, Making Sense of Video Streaming Protocols, Linkedin. (Aug. 11, 2016), https://www.linkedin.com/pulse/making-sense-video-streaming-protocols-dr-brijesh-kumar/ ("Kumar"), 7 pages (Brijesh_Kumar_Making_Sense_2016).
Charles Marion & Julien Jomier, Real-time Collaborative Scientific WebGL Visualization with Websocket, Ass'n Computing Machinery, Aug. 4-5, 2012 ("Marion"), 4 pages (CHARLES_MARION_Real-time_2012).
Christopher Mueller, MEPG-DASH vs. *Apple HLS vs. Microsoft Smooth Streaming* vs. *Adobe Hds, Bitmovin*. (Mar. 29, 2015), https://bitmovin.com/mpeg-dash-vs-apple-hls-vs-microsoft-smooth-streaming-vs-adobe-hds/ ("Mueller"), 10 pages (Christopher_Mueller_MEPG-DASH_Bitmovin_2015).
Curriculum Vitae of Dr. Stephen Wicker, PhD., 38 pages (Wicker_CV_38p).
Declaration of Dr. Stephen Wicker, Ph.D., signed Aug. 22, 2024, in IPR2024-01305, 499 pages (WIcker_Dec_IPR2024-01305_499p).
Declaration of Dr. Stephen Wicker, Ph.D., signed Aug. 22, 2024, in IPR2024-01307, 499 pages (WIcker_Dec_IPR2024-01307_499p).
Declaration of Dr. Stephen Wicker, Ph.D., signed Aug. 22, 2024, in IPR2024-01308, 499 pages (WIcker_Dec_IPR2024-01308_499p).
Declaration of Dr. Stephen Wicker, Ph.D., signed Aug. 22, 2024, in IPR2024-01309, 499 pages (WIcker_Dec_IPR2024-01309_499p).
Declaration of Dr. Stephen Wicker, Ph.D., signed Aug. 22, 2024, in IPR2024-01310, 499 pages (WIcker_Dec_IPR2024-01310_499p).
Ilya Grigorik, WebRTC, in High Performance Browser Networking, (2013) ("Grigorik"), 20 pages (Ilya Grigorik, WebRTC, in High Performance Browser Networking, (2013) ("Grigorik"), 20 pages (Ilya_Grigorik_WebRTC_High_Performance_2013).
Declaration of Dr. Stephen Wicker, Ph.D., signed Aug. 22, 2024, in IPR2024-01311, 499 pages (WIcker_Dec_IPR2024-01311_499p).
Genius Sports' proposed Sotera Stipulation to SportsCastr, 2 pages (Genius_propose_Sotera_Stip).
Hardeep Singh Parmar & Michael C. Thornburgh, Adobe's Real Time Messaging Protocol, Veriskope. (Dec. 21, 2002) ("Parmer"), 52 pages (Hardeep_Singh_Parmar_Adobe_Veriskope_2002).
Henning Schulzrinne et al., RTP: A Transport Protocol for Real-Time-Applications, Data Tracker (Jan. 1996), https://datatracker.ietf.org/doc/html/rfc1889 ("Schulzrinne"), 75 pages (Henning_Schulzrinne_RTP_1996).
Hisayuuki Ohmata et al., Hybridcast: A new media experience by integration of broadcasting and broadband, 2013 ITU Kaleidoscope: Building Sustainable Communities, Apr. 2013 ("Ohmata"), 8 pages (Hisayuuki_Ohmata_Hybridcast_2013).
Ian Fette & Alexey Melnikov, The WebSocket Protocol, RFC Editor. (Dec. 2016), https://www.rfc-editor.org/rfc/rfc6455. html, 71 pages (Ian_Fette_WebSocket_RFC_2016).
James F. Kurose & Keith W. Ross, Application Layer, in Computer Networking A Top-Down Approach (Michael Hirsch et al. eds., 4th ed. 2012) ("Kurose"), 5 pages (James_F_Kurose_2012).
Jan Newmarch, Introduction to Stream Control Transmission Protocol, Linux J. (Sep. 1, 2007), https://www.linuxjournal.com/article/9748 ("Newmarch"), 12 pages (Jan_Newmarch_Introduction_LINUX_2007).
Jan Ozer, Upstream, Justin.tv, Livestream, and Bambuser: Streaming Unplugged, Streaming Media Magazine., Jun.- Jul. 2012 ("Ozer"), 18 pages (Jan_Ozer_Upstream_2012).
John Dilley et al., Globally Distributed Content Delivery, IEEE Internet Computing (Sep.-Oct. 2002), 9 pages (john_Dilley_Globally_distributed_2002).
Kyung-Hoe Kim et al., A transmission control SCTP for real-time multimedia streaming, 54 Computer Networks 1418, (2010) ("Kim"), 8 pages (KYUNG_KIM_transmission_54_Computer_2010).
Lucian Popa et al., HTTP as the narrow waist of the future internet, 2010 Acm Sigcomm Workshop, Oct. 2013 ("Popa"), 6 pages (Lucian_Popa_HTTP_ACM_2013).
Lydon Ong, John Yoakum &, An Introduction to the Stream Control Transmission Protocol (SCTP), Datatracker. (May 2002), https://datatracker.ietf.org/doc/rfc3286/ ("Ong"), 10 pages (Lydon_Ong_Introduction_Datatracker_2002).
Marcin Warczygłowa, Real-time Web Application with Websockets and Vert.X (Nov. 10, 2015), https://blog.allegro.tech/2015/11/real-

(56) **References Cited**

OTHER PUBLICATIONS

time-web-application-with-websockets-and-vert-x.html, 14 pages (Marcin_Warczygłowa_Real- time_Web_blog-allegro-tech_2015).
Mark Sweney, First ads appear on YouTube Clips, Guardian, Aug. 22, 2007 ("Sweney"), 6 pages (Mark_Sweney_First- ads_Guardian_2007).
Petition for Inter Partes Review of U.S. Pat. No. 10,425,697 (IPR2024-01308) dated Aug. 22, 2024, 97 pages (Petition-IPR2024-01308).
Petition for Inter Partes Review of U.S. Pat. No. 10,425,697 (IPR2024-01309) dated Aug. 22, 2024, 101 pages (Petition-IPR2024-01309).
Petition for Inter Partes Review of U.S. Pat. No. 10,805,687 (IPR2024-01305) dated Aug. 22, 2024, 99 pages (Petition-IPR2024-01305).
Petition for Inter Partes Review of U.S. Pat. No. 10,805,687 (IPR2024-01307) dated Aug. 22, 2024, 101 pages (Petition-IPR2024-01307).
Petition for Inter Partes Review of U.S. Pat. No. 11,039,218 (IPR2024-01310) dated Aug. 23, 2024, 100 pages (Petition-IPR2024-01310).
Petition for Inter Partes Review of U.S. Pat. No. 11,039,218 (IPR2024-01311) dated Aug. 23, 2024, 103 pages (Petition-IPR2024-01311).
Randall Stewart & Chris Metz, SCTP: new transport protocol for TCP/IP, 5 IEEE 64, (2001) ("Stewart"), 6 pages (Randall_Stewart_SCTP_IEEE_2001).
Richard Sandomir, The Innovation That Grew and Grew, N.Y. Times, Jun. 12, 2014 ("Sandomir"), 3 pages (Richard_Sandomir_Innovation_NY_Times_2014).
Sam Dutton, Get started with WebRTC, web.dev. (Jul. 23, 2012), https://web.dev/articles/webrtc-basics ("Dutton"), 27 pages (Sam_Dutton_Get_started_w_WebRTC_2012).
SportsCastr's Complaint for Patent Infringement against Genius Sports, dated Oct. 5, 2023, 123 pages (SGIC_Complaint_Genius).
SportsCastr's Preliminary Infringement Contentions dated Jan. 18, 2024, 638 pages (SGIC_Prelim_Infring_Cont).
SportsCastr's Response to Genius Sports' First Set of Interrogatories dated May 29, 2024, 53 pages (SGIC_RESP- to_Ints).
Stefan Saroiu et al., An analysis of Internet Content Delivery Systems, Acm Sigops Operating System Review, vol. 36 (Dec. 31, 2002), https://doi.org/10.1145/844128.84415 ("Saroiu"), 13 pages (Stefan_Saroiu_analysis_ACM_2002).
Victoria Pimentel & Bradford G. Nickerson, Communicating and Displaying Real-Time Data with WebSocket, 16 IEEE (2012) ("Pimentel"), 9 pages (Victoria_Pimentel _Communicating_IEEE_2012).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-23, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 189 pages (SRAD-EX-A-23-WebSocket).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-24, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 200 pages (SRAD-EX-A-24-AppleHLS).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-25, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 331 pages (SRAD-EX-A-25-Betstream).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-26, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 79 pages (SRAD-EX-A-26-FacebookLive).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-27, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 90 pages (SRAD-EX-A-27-MLBAtBat).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit A-28, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 78 pages (SRAD-EX-A-28-OneTwoSee).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-01, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 158 pages (SRAD-EX-B-01-Lu).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-02, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 302 pages (SRAD-EX-B-02-Dukes).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-03, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 143 pages (SRAD-EX-B-03-Mandyam).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-04, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 168 pages (SRAD-EX-B-04-Cromarty).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-05, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 171 pages (SRAD-EX-B-05-Cooper).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-06, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 190 pages (SRAD-EX-B-06-Basso).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-07, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 166 pages (SRAD-EX-B-07-Sullivan).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-08, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 93 pages (SRAD-EX-B-08-Bota).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-09, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 56 pages (SRAD-EX-B-09-Dengler).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-10, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 73 pages (SRAD-EX-B-10-Ma).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-11, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 56 pages (SRAD-EX-B-11-Mukherjee).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-12, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 127 pages (SRAD-EX-B-12-Ellis).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-13, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 219 pages (SRAD-EX-B-13-Herzog).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-14, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 222 pages (SRAD-EX-B-14-Mitic).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-15, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 107 pages (SRAD-EX-B-15-Spivey).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-16, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 692 pages (SRAD-EX-B-16-Levinson).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-17, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 284 pages (SRAD-EX-B-17-Park).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-18, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr*

(56) **References Cited**

OTHER PUBLICATIONS

Inc. (dba Panda Interactive) v. *Sportradar Group AG and Sportradar AG*, 193 pages (SRAD-EX-B-18-Abulikemu).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-20, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 158 pages (SRAD-EX-B-20-Kaazing).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-21, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 244 pages (SRAD-EX-B-21-WebRTC).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-22, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 288 pages (SRAD-Ex-B-22-RTP).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-23, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 190 pages (SRAD-EX-B-23-WebSocket).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-24, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 201 pages (SRAD-EX-B-24-AppleHLS).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-25, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 302 pages (SRAD-EX-B-25-Betstream).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-26, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 78 pages (SRAD-EX-B-26-FacebookLive).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-27, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 90 pages (SRAD-EX-B-27-MLBAtBat).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit B-28, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 78 pages (SRAD-EX-B-28-OneTwoSee).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-01, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 126 pages (SRAD-EX-C-01-Lu).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-02, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 264 pages (SRAD-EX-C-02-Dukes).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-03, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 111 pages (SRAD-EX-C-03-Mandyam).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-04, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 133 pages (SRAD-EX-C-04-Cromarty).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-05, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 96 pages (SRAD-EX-C-05-Cooper).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-06, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 161 pages (SRAD-EX-C-06-Basso).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-07, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 138 pages (SRAD-EX-C-07-Sullivan).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-08, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 97 pages (SRAD-EX-C-08-Bota).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-09, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 72 pages (SRAD-EX-C-09-Dengler).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-10, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 85 pages (SRAD-EX-C-10-Ma).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-11, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 59 pages (SRAD-EX-C-11-Mukherjee).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-12, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 87 pages (SRAD-EX-C-12-Ellis).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-13, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 125 pages (SRAD-EX-C-13-Herzog).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-14, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 275 pages (SRAD-EX-C-14-Mitic).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-15, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 109 pages (SRAD-EX-C-15-Spivey).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-16, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 504 pages (SRAD-EX-C-16-Levinson).
Sportradar Group AG and Sportradar AG's Invalidity Contentions Exhibit C-17, May 6, 2024, Case No. 2:23-cv-00472-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Sportradar Group AG and Sportradar AG*, 213 pages (SRAD-EX-C-17-Park).
56 Bit Sportradar Case Study, Sportradar Time System 2022 accessed at https://www.56bit.com/case-studies/sportradar-time-system on Feb. 16, 2023, 5 pages https://info.maestro.io/analyze-and-optimize on Apr. 21, 2021. 2 pages.
Analyze & Optimize. Maestro 2020. Accessed at pages.
API Feed Overview and Documentation, Jan. 31, 2022, GeniusSport Support Centre, 3 pages.
Around-the-clock action for you sportsbook. BetGenius Brochure. Accessed at https://geniussports.com/sportsbook/content/streaming/ on Mar. 19, 2021. 1 page.
Betbuilder Architecture, GeniusSports, Oct. 13, 2022, 5 pages.
BetBuilder evolves to provide the ultimate pre-match experience, GeniusSports Website Sep. 18, 2019, 3 pages.
Betbuilder Integration—Prebuilt bets GeniusSports Wiki, May 25, 2021, 2 pages.
Betbuilder Integration, GeniusSports, Oct. 13, 2022, 1 page.
Betgenius Adds Live Streaming to Tipsport Deal, Gaming Intelligence, Sep. 23, 2020, 1 page.
Betgenius Launches Live Streaming Service for Sportsbooks, Genius Sports, Oct. 8, 2019, 3 pages.
Betgenius Streaming, Bring your sportsbook to Life-Right Around the Clock, Dec. 3, 2020, 2 pages.
Betradar 10 Benefits of our New Audio Visual Player—Jul. 10, 2019, accessed at https://www.betradar.com/news-archive/the-top-10-benefits-of-our-new-audio-visual-player/ on Feb. 24, 2023, 1 page.
Betradar Audio-Visual Brochure 2022, 8 pages.
Betradar Audio-Visual Brochure 2023, 9 pages.
Betradar Brochure—MTS SDK .Net Integration Guide—Oct. 2020 accessed at https://sdk.sportradar.com/mts/net/file?g=ab6006f6-f554-482f-92db-26af416eec92 on Feb. 17, 2023, 14 pages.
Betradar Brochure Live Scouting & Live Odds Services, 24/7 Live Betting Coverage from the market leader, 2015, 12 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Betradar Brochure Live-Data Service 2022 accessed at https://www.betradar.com/wp-content/uploads/sites/4/2022/11/Live-Data-Brochure-2022.pdf, 5 pages.
Betradar Brochure MTS SDK Java Integration Guide—May 2019 accessed at https://sdk.sportradar.com/mts/java/file? g=998cd241-1182-4c30-9f54-697e6cadad7b on Feb. 18, 2023, 14 pages.
Betradar CTRL User Manual, Jul. 2019 accessed at https://insidectrl.com/wp-content/uploads/2019/09/Betradar_Ctrl_Manual_17_July_2019.pdf, on Feb. 16, 2023, 88 pages.
Betradar CTRL User Manual, Mar. 2021 access at https://insidectrl.com/help-manual/ on Feb. 16, 2023, 102 pages.
Betradar Live Channel Online, Dec. 31, 2022, 3 page.
Betradar Live Channel Trading, Dec. 31, 2022, 2 page.
Betradar Live Data Service-Efficiently Operate Your Own In-running trading, Dec. 30, 2022, accessed at https://www.betradar.com/betting-services/live-data-service/ on Feb. 16, 2023, 3 page.
Betradar Live Streaming Brochure 2019, accessed at https://betradar.com/wp-content/uploads/sites/4/2015/07/ Live_Channel_Brochure_2019.pdf on Feb. 24, 2023, 7 pages.
Betradar Live Streaming Brochure 2021. Accessed at https://www.betradar.com/wp-content/uploads/sites/4/2014/11/Betradar-Live-Streaming-Brochure.pdf on Apr. 29, 2021. 7 pages.
Betradar Market Leading Sports Betting Product Brochure 2021. Accessed at https://www.betradar.com/wp-content/uploads/sites/4/2021/02/Betradar-Product-Brochure-2021.pdf on Apr. 29, 2021. 24 pages.
Betradar Market-leading Sports Betting Services Product Brochure 2022, 23 pages.
Betradar Product Brochure 2015, 27 pages.
Betradar Product Brochure 2016, 29 pages.
Betradar Product Brochure 2018, 15 pages.
Betradar Product Brochure 2019, 36 pages.
Betradar Top Reasons to Use the New Av Api, Jul. 31, 2019, accessed at https://www.betradar.com/news-archive/61-reasons-to-use-the-new-av-api/ on Feb. 24, 2023, 2 pages.
Betradar Unified Odds Feed—Quick Start Guide, May 16, 2018, accessed at https://insidectrl.com/wp-content/uploads/2018/05/UOF-Start-Guide.pdf on Apr. 13, 2023, 5 pages.
Betradar Unified Odds Feed—Quick Start Guide, May 30, 2022, accessed at https://iodocs.betradar.com/unifiedsdk/UOF_Start_Guide.pdf on Feb. 17, 2023, 6 pages.
Betradar Unified Odds-Integration Information for Development, Mar. 27, 2019 accessed at https://iodocs.betradar.com/unifiedsdk/Betradar_Unified-Odds_Developer_Integration.pdf on Feb. 16, 2023, 137 pages.
Betradar Unified Odds-Integration Information for Development, Oct. 11, 2017 accessed at https://insidectrl.com/wp-content/uploads/2017/10/2017-10-11_Betradar_Unified-Odds_Developer_Integration-1.pdf on Feb. 16, 2023, 80 pages.
Betradar. Sportradar. Accessed at https://www.betradar.com/ on Apr. 21, 2021. 2 pages.
Betradar's Live Channels. Betrader Brochure 2017, 7 pages.
Boolabus, Bring it to life with Betstream, Feb. 2011, 22 pages.
Common Match State Platform APIs GeniusSports Wiki Jul. 5, 2022, 2 pages.
Customize & Control. Maestro 2020. Accessed at https://info.maestro.io/customize-and-control on Apr. 21, 2021. 2 pages.
Empson, Baseball's Digital Trifecta and How America's Pastime is Setting the Pace for Sports Online. TechCrunch Nov. 4, 2013. Accessed at https://techcrunch.com/2013/11/04/baseballs-digital-trifecta-and-how-americas-pastime-is-setting-the-pace-for-sports-online/. 10 pages.
Engage & Monetize. Maestro 2020. Accessed at https://info.maestro.io/engage-and-monetize on Apr. 21, 2021. 3 pages.
Esports—Take your fan experience to the next level and build community. Maestro 2020. Accessed at https://info.maestro.io/esports on Apr. 21, 2021. 4 pages.
Gammastack-Betradar Integration Architecture, Dec. 29, 2022 accessed at https://www.gammastack.com/betradar-integration/ on Feb. 16, 2023, 7 pages.
GammaStack-Betradar Integration Services, Betradar Software, Dec. 29, 2022, accessed at https://www.gammastack.com/betradar-integration-services/ on Feb. 17, 2023, 6 pages.
Genius Play Integration Guide, Drop and Play, GeniusSports Wike Website, Jul. 19, 2021, 7 pages.
Genius Sports Expands Partnership with NFL to Provide Watch & Bet Video Streams For the 2022 Season, Starting with Caesars Entertainment, BusinessWire, Dec. 5, 2022, 2 pages.
Genius Sports Group Analyst Day Presentation Jan. 2021. Accessed at https://s27.q4cdn.com/552951210/files/doc_downloads/GSG_Analyst_Day_Presentation_January_2021_vFF_2.pdf on Apr. 29, 2021. 52 pages.
Genius Sports Group Investor Presentation Oct. 2020. Accessed at https://news.geniussports.com/wp-content/uploads/2020/10/GSG-Investor-Presentation.pdf on Apr. 29, 2021. 29 pages.
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit D-4, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 25 pages (Ex_D-4_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-1, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD, 212 pages* (Ex_E-01_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-10, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 311 pages (Ex_E-10_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-11, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 372 pages (Ex_E-11_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-12, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 85 pages (Ex_E-12_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-13, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 114 pages (Ex_E-13_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-14, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 140 pages (Ex_E-14_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-15, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 205 pages (Ex_E-15_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-16, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 247 pages (Ex_E-16_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-17, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 423 pages (Ex_E-17_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-18, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 408 pages (Ex_E-18_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-19, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 177 pages (Ex_E-19_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-2, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 453 pages (Ex_E-02_088_npl).
Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-20, Jul. 29, 2024, Case No.

(56) **References Cited**

OTHER PUBLICATIONS

2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 233 pages (Ex_E-20_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-21, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 204 pages (Ex_E-21_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-22, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 181 pages (Ex_E-22_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-23, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 192 pages (Ex_E-23_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-24, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 184 pages (Ex_E-24_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-25, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 236 pages (Ex_E-25_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-26, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 329 pages (Ex_E-26_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-3, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 245 pages (Ex_E-03_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-4, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 602 pages (Ex_E-04_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-5, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 76 pages (Ex_E-05_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-6, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 183 pages (Ex_E-06_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-7, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 53 pages (Ex_E-07_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-8, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 68 pages (Ex_E-08_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions Exhibit E-9, Jul. 29, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 237 pages (Ex_E-09_088_npl).

Genius Sports LTD's First Supplemental Invalidity and Subject-Matter Eligibility Contentions served on Jul. 29, 2024, in U.S. District Court, Eastern Division of Texas Marshall Division, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 58 pages (2024-07-29_Genius_Supp_Cont_npl).

Real-Time Second Screen Connectivity, http:///www.kaazing.com/solutions/real-time-second-screen/connectivity/ (Nov. 10, 2016) (GS_00012474) (Second Screen Connectivity), 1 page.

Arini, St, Mt, ch 6 Multimedia Distribution, https://www.slideserve.com/tassos/ch-6-multimedia-distribution (Oct. 11, 2024), (ARINI_CH-6_2014) 37 pages.

Brian Snelson, CloudFlare Now Supports WebSockets (Aug. 5, 2014), (Brian_Snelson_Cloudflare_2014) 14 pages.

Christian Hessleman, et al., "Sharing Enriched Multimedia Experiences accross Heterogeneous Network Infrastructures" IEEE Communications Magazine 48.6 (2010): 54-65, (Hessleman_et_al_Sharing_IEEE_2010).

EX-2001 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 IPRs 2024-01305, 01307, 01310 and 01311, Declaration of Rubin, Ph.D., 167 pages (POPR-Ex-2001).

EX-2002 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Curriculum vitae of Aviel Rubin, Ph.D., 24 pages (POPR-Ex-2002).

EX-2003 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Joint Claim Construction Statement, Dkt. 69, 24 pages (POPR-Ex-2003).

EX-2004 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Declaration of Dr. Stephen Wicker, Sep. 13, 2024, 121 pages (POPR-Ex-2004_QC) 7 parts.

EX-2005 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Yahoo Finance Tanner Simkins, Apr. 20, 2017, 3 pages (POPR-Ex-2005).

EX-2006 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—NBA.com, Steve Ashbumer Apr. 27, 2017 Article, 5 pages (POPR-Ex-2006).

EX-2007 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Sportstechie, Robert Roble, Apr. 22, 2017, 9 pages (POPR-Ex-2007).

EX-2008 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Sports Tech Tokyo Award; 2019 Yahoo! Sports Tech Awards; Sports Business Awards, 6 pages (POPR-Ex-2008_QC).

EX-2009 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Various Media Reports on the Patented Invention, 28 pages (POPR-Ex-2009_QC).

EX-2010 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—CNBC, Dec. 20, 2018, 1 page (POPR-Ex-2010_QC).

EX-2011 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01211—https://sportscastr.com/about, 3 pages (POPR-Ex-2011).

EX-2012 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—https://pandainteractive.com/patents, 9 pages (POPR-Ex-2012_QC).

EX-2013 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—https://sportscastr.com, 2 pages (POPR-Ex-2013).

EX-2014 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—https://studiopanda.live/, 1 page (POPR-Ex-2014).

EX-2015 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Genius Sports Feb. 24, 2018 Email, 3 pages (POPR-Ex-2015).

EX-2016 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Genius Sports May 7, 2018 Email, 3 pages (POPR-Ex-2016).

EX-2017 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—'687 Patent Genius Sports Infringement Claim Chart, 50 pages (POPR-Ex-2017).

EX-2018 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—https://www.opensourceforu.com/2015/03/a-guide-to-using-raw-sockets, 10 pages (POPR-Ex-2018).

EX-2019 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and

(56) **References Cited**

OTHER PUBLICATIONS

01311—https://learn.microsoft.com/en-US/windows/win32/winsock/tcp-ip-raw-sockets-2, 5 pages (POPR-Ex-2019).
EX-2020 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Oracle documentation in 2011: Anatomy of the Client/Server Model. https://docs.oracle.com/cd/E13203_01/tuxedo/tux80/atmi/intbas3.htm, 3 pages (POPR-Ex-2020).
EX-2021 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Peng, Gang. "CDN: Content distribution network." arXiv preprint cs/0411069 (2004) https://arxiv.org/pdf/cs/0411069, 26 pages (POPR-Ex-2021.
EX-2022 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—The History of Streaming Media. https://boyetblog.s3.amazonaws.com/PCPlus/324.Streaming.pdf, 3 pages (POPR-Ex-2022).
EX-2023 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—RFC 2326: Real Time Streaming Protocol (RTSP) https://www.ietf.org/rfc/ffc2326.txt, 72 pages (POPR-Ex-2023).
EX-2024 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Adobe's Real Time Messaging Protocol https://rtmo.veriskooe.com/pdf/rtmp specification 1.0.pdf, 52 pages (POPR-Ex-2024).
EX-2025 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—WebRTC 1.0: Real time Communication Between Browsers https://www.w3.org/TR/2011/WD-webrtc-20111027/, 28 pages (POPR-Ex-2025).
EX-2026 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—HTTP Live Streaming: draft-pantos-http-live-streaming-01 https://datatracker.ietf.org/doc/html/draft-pantos-http-live-streaming-01, 17 pages (POPR-Ex-2026).
EX-2027 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Microsoft To Add Smooth Streaming in IIS7 https://virtualizationreview.com/articles/2008/11/04/microsoft-to-add-smooth-streaming-in-iis7.aspx, 4 pages (POPR-Ex-2027).
EX-2028 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—HTTP Dynamic Streaming. (Archived May 14, 2010) https://web.archive.org/web/20100514170442/http://www.adobe.com/products/httpdynamicstreaming/, 2 pages (POPR-Ex-2028).
EX-2029 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—MPEG ratifies its draft standard for DASH. (Archived Dec. 14, 2011) https://web.archive.org/web/20111214104626/http://mpeg.chiari glione.org/meetings/geneva11-1/Geneva_press.htm, 4 pages (POPR- Ex-2029).
EX-2030 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Mogulus Rebrands With A Killer Domain: Livestream.com https://techcrunch.com/2009/05/18/mogulus-gets-a-killer-domain-for-rebranding-livestreamcom/, 6 pages (POPR-Ex-2030).
EX-2031 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310. And 01311—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/, 6 pages (POPR-Ex-2031).
EX-2032 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—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/, 4 pages (POPR-Ex-2032).
EX-2033 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Twitch Engineering: An Introduction and Overview, https://blog.twitch.tv/en/2015/12/18/twitch-engineering-an-introduction-and-overview-a23917b71a25/, 8 pages (POPR-Ex-2033).
EX-2034 to Patent Owners Preliminary Responses dated Dec. 19, 2024 and Dec. 20, 2024 in IPRs 2024-01305, 01307, 01310 and 01311—Proceedings of the 25th ACM workshop on network and operating systems support for digital audio and video, pp. 55-60, 2015 https://arxiv.org/ndf/1502.04666, 7 pages (POPR-Ex-2034_QC).
Francesco Cricri, Sujeet Mate, Igor D.D. Cucio, and Moncef Gabbouj, "Mobile and Interactive Social Television—A Virtual TV Room" 2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops. IEEE, 2009, 7 pages (Francesco_Cricri_Mobile_IEEE_2009).
Sports' 2nd Supp. Invalidity Contentions, Ex A-01S, Dec. 9, 2024, 20 pages (GEN-Ex-A-01S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-02S, Dec. 9, 2024, 20 pages (GEN-Ex-A-02S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-03S, Dec. 9, 2024,20 pages (GEN-Ex-A-03S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-04S, Dec. 9, 2024, 20 pages (GEN-Ex-A-04S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-05S, Dec. 9, 2024, 20 pages (GEN-Ex-A-05S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-06S. Dec. 9, 2024, 20 pages (GEN-Ex-A-06S_12-9-24).
2nd Supp. Invalidity Contentions, Ex A-07S, Dec. 9, 2024, 20 pages (GEN-Ex-A-07S_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-08S, Dec. 9, 2024, 21 pages (GEN-Ex-A-08S_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-09S, Dec. 9, 2024, 20 pages (GEN-Ex-A-09S_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-10S. Dec. 9, 2024, 20 pages (GEN-Ex-A-10S_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-11S, Dec. 9, 2024, 20 pages (GEN-Ex-A-11S_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-12S, Dec. 9, 2024, 21 pages (GEN-Ex-A-12S_12-9-24).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-9, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 326 pages (Ex_C-09_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit D-1, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 22 pages (Ex_D-01_687_Patent_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit D-2, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 25 pages (Ex_D-02_697_Patent_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit D-3, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 20 pages (Ex_D-03_218_Patent_npl).
IPR 2025-00251 and 00252 (Dec. 6, 2024) Exhibit 1001, "Wicker Dec" 223 pages (IPR2025-00251_EX_1001_223_p_12-06-2024_npl).
IPR 2025-00251 and 00252 (Dec. 6, 2024) Exhibit 1002, "Wicker CV" 38 pages (IPR2025-00251_EX_1002_38_p_12-06-2024_npl).
IPR 2025-00251 and 00252 (Dec. 6, 2024) Exhibit 1013, "Vidal Memo" 9 pages (IPR2025-00251_EX_1013_57_p_12-06-2024_npl).
IPR 2025-00251 and 00252 (Dec. 6, 2024) Exhibit 1057, 57 pages (IPR2025-00251_EX_1057_57_p_12-06-2024_npl).
IPR 2025-00251 and 00252 (Dec. 6, 2024) Exhibit 1058, "ARINI" 37 pages (IPR2025-00251_EX_1058_37_p_12-06-2024_npl).
IPR 2025-00251 and 00252 (Dec. 6, 2024) Exhibit 1059, "Sportscastr 2nd Am. Comp" 173 pages (IPR2025-00251_EX_1059_173_p).
James F. Kurose and Keither W. Ross, Computer Networking: A Top-Down Approach (Michale Hirsch et al. eds, 6th ed 2013), 889 pages.
Lancerica et al., MITv—A solution for an Interactive TV based on IP Multicast over Satellite, 2004 IEEE Int'l Conf. on Multimedia

(56) **References Cited**

OTHER PUBLICATIONS

and Expo (ICME) (Jun. 27-30, 2004) https://leeexplore.jeee.org/document/1394696, 4 pages (Lancerica_et_al_MITv_IEEE_2004).
Luigi Lo Jacono and Silvia Santano Guillen, Web-native, Video Live Streaming, WEB 2014: The Second International Conference on Building and Exploring Web Based Environments, Apr. 2014, 7 pages.
Petition for Inter Partes Review of U.S. Pat. No. 11,871,088 (IPR 2025-00251), filed Dec. 6, 2024, 106 pages (IPR2025-00251_Petition_12-Jun. 2024).
Petition for Inter Partes Review of U.S. Pat. No. 11,871,088 (IPR 2025-00252), filed Dec. 6, 2024, 1-6 pages (IPR2024-00252_Petition_12-Jun. 2024) 104 pages.
Petitioners' Reasons for Parallel Petitioners and Petition Ranking, (IPR 2025-00251), filed Dec. 6, 2024, 1-6 pages (IPR2025-00251_Pet_Reasons_Parallel_Petitions_12-Jun. 2024).
Petitioners' Reasons for Parallel Petitions and Petition Ranking in IPR 2025-00252, filed Dec. 6, 2024, 1-6 pages (IPR2025-00252_Peti_Reasons_Parallel_Petitions_12-Jun. 2024).
Sportradar AG and Sportradar Group AG's Amended Invalidity Contentions in United States District Court for Eastern District of Texas, Case No. 2:23-cv-00472, served Dec. 9, 2024, 80 pages (SRAD_Am_Cont_12-Sep. 2024).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-01, Dec. 9, 2024, 213 pages (SRAD_Ex- A-01_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-02, Dec. 9, 2024, 444 pages (SRAD_Ex- A-02_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-03, Dec. 9, 2024, 600 pages (SRAD_Ex- A-03_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-04, Dec. 9, 2024, 309 pages (SRAD_Ex- A-04_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-05, Dec. 9, 2024, 372 pages (SRAD_Ex- A-05_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-06, Dec. 9, 2024, 330 pages (SRAD_Ex- A-06_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-07, Dec. 9, 2024, 309 pages (SRAD_Ex- A-07_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-08, Dec. 9, 2024, 148 pages (SRAD_Ex- A-08_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-09, Dec. 9, 2024, 111 pages (SRAD_Ex- A-09_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-10, Dec. 9, 2024, 128 pages (SRAD_Ex- A-10_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-11, Dec. 9, 2024, 202 pages (SRAD_Ex- A-11_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-12, Dec. 9, 2024, 312 pages (SRAD_Ex- A-12_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-13, Dec. 9, 2024, 282 pages (SRAD_Ex- A-13_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-14, Dec. 9, 2024, 323 pages (SRAD_Ex- A-14_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-15, Dec. 9, 2024, 197 pages (SRAD_Ex- A-15_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-16, Dec. 9, 2024, 978 pages (SRAD_Ex- A-16_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-17, Dec. 9, 2024, 567 pages (SRAD_Ex- A-17_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-18, Dec. 9, 2024, 320 pages (SRAD_Ex- A-18_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-20, Dec. 9, 2024, 265 pages (SRAD_Ex- A-20_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-21, Dec. 9, 2024, 429 pages (SRAD_Ex- A-21_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-22, Dec. 9, 2024, 690 pages (SRAD_Ex- A-22_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-23, Dec. 9, 2024, 326 pages (SRAD_Ex- A-23_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-24, Dec. 9, 2024, 490 pages (SRAD_Ex- A-24_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-25, Dec. 9, 2024, 614 pages (SRAD_Ex- A-25_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-26, Dec. 9, 2024, 265 pages (SRAD_Ex- A-26_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-27, Dec. 9, 2024, 175 pages (SRAD_Ex- A-27_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-28, Dec. 9, 2024, 268 pages (SRAD_Ex- A-28_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-30, Dec. 9, 2024, 639 pages (SRAD_Ex- A-30_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-31, Dec. 9, 2024, 443 pages (SRAD_Ex- A-31_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-32, Dec. 9, 2024, 269 pages (SRAD_Ex- A-32_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-33, Dec. 9, 2024, 335 pages (SRAD_Ex- A-33_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-34, Dec. 9, 2024, 381 pages (SRAD_Ex- A-34_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-32, Dec. 9, 2024, 139 pages (GEN-Ex-B-32_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-33, Dec. 9, 2024, 292 pages (GEN-Ex-B-33_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-34, Dec. 9, 2024, 212 pages (GEN-Ex-B-34_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-35, Dec. 9, 2024, 117 pages (GEN-Ex-B-35_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex C-15, Dec. 9, 2024, 50 pages (GEN-Ex-C-15_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex C-16, Dec. 9, 2024, 74 pages (GEN-Ex-C-16_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex C-18, Dec. 9, 2024, 50 pages (GEN-Ex-C-18_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex C-20, Dec. 9, 2024, 52 pages (GEN-Ex-C-20_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-21, Dec. 9, 2024, 45 pages (GEN-Ex-C-21_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-22, Dec. 9, 2024, 48 pages (GEN-Ex-C-22_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-23, Dec. 9, 2024, 44 pages (GEN-Ex-C-23_12-9-24).

(56) **References Cited**

OTHER PUBLICATIONS

Genius Sports' 2nd Supp. Invalidity Contentions, Ex C-24, Dec. 9, 2024, 45 pages (GEN-Ex-C-24_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-25, Dec. 9, 2024, 67 pages (GEN-Ex-C-25_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-26, Dec. 9, 2024, 64 pages (GEN-Ex-C-26_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-27, Dec. 9, 2024, 94 pages (GEN-Ex-C-27_12-9-24)(2 parts).
Sports' 2nd Supp. Invalidity Contentions, Ex C-28, Dec. 9, 2024, 42 pages (GEN-Ex-C-28_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-29, Dec. 9, 2024, 86 pages (GEN-Ex-C-29_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-30, Dec. 9, 2024, 112 pages (GEN-Ex-C-30_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-31, Dec. 9, 2024, 84 pages (GEN-Ex-C-31_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-32, Dec. 9, 2024, 52 pages (GEN-Ex-C-32_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-33, Dec. 9, 2024, 75 pages (GEN-Ex-C-33_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-34, Dec. 9, 2024, 80 pages (GEN-Ex-C-34_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex C-35, Dec. 9, 2024, 37 pages (GEN-Ex-C-35_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex D-1, Dec. 9, 2024, 26 pages (GEN-Ex-D-1_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex D-2, Dec. 9, 2024, 30 pages (GEN-Ex-D-2_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex D-3, Dec. 9, 2024, 23 pages (GEN-Ex-D-3_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex D-4, Dec. 9, 2024, 27 pages (GEN-Ex-D-4_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex E-18, Dec. 9, 2024, 408 pages (GEN-Ex-E-18_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex E-27, Dec. 9, 2024, 62 pages (GEN-Ex-E-27_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex E-28, Dec. 9, 2024, 42 pages (GEN-Ex-E-28_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex E-29, Dec. 9, 2024, 54 pages (GEN-Ex-E-29_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex E-30, Dec. 9, 2024, 86 pages (GEN-Ex-E-30_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex E-31, Dec. 9, 2024, 85 pages (GEN-Ex-E-31_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex E-32, Dec. 9, 2024, 46 pages (GEN-Ex-E-32_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex E-33, Dec. 9, 2024, 49 pages (GEN-Ex-E-33_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex E-34, Dec. 9, 2024, 74 pages (GEN-Ex-E-34_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex E-35, Dec. 9, 2024, 42 pages (GEN-Ex-E-35_12-9-24).
Genius Sports' 2nd Supplemental Invalidity and Subject Matter Eligibility Contentions, filed in United States District Court for Eastern District of Texas, Civil Action Nos. 2:23-cv-00472 and 2:23-cv-00471) served Dec. 9, 2024, 74 pages.
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-10, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 980 pages (Ex_C-10_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-11, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 777 pages (Ex_C-11_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-12, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 1206 pages (Ex_C-12_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-13, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 321 pages (Ex_C-13_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-14, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 401 pages (Ex_C-14_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-2, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 925 pages (Ex_C-02_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-3, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 627 pages (Ex_C-03_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-4, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 1339 pages (Ex_C-04_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-5, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 458 pages (Ex_C-05_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-6, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 345 pages (Ex_C-06_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-7, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 438 pages (Ex_C-07_218_npl).
Genius Sports LTD's Initial Invalidity and Subject-Matter Eligibility Contentions Exhibit C-8, Mar. 14, 2024, Case No. 2:23-cv-00471-JRG, *Sportscastr Inc. (dba Panda Interactive)* v. *Genius Sports LTD*, 500 pages (Ex_C-08_218_npl).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-35, Dec. 9, 2024, 369 pages (SRAD_Ex- A-35_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-36, Dec. 9, 2024, 119 pages (SRAD_Ex- A-36_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-37, Dec. 9, 2024, 400 pages (SRAD_Ex- A-37_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-38, Dec. 9, 2024, 15 pages (SRAD_Ex- A-38_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-39, Dec. 9, 2024, 31 pages (SRAD_Ex- A-39_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex A-40, Dec. 9, 2024, 29 pages (SRAD_Ex- A-40_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-01, Dec. 9, 2024, 211 pages (SRAD_Ex- B-01_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-02, Dec. 9, 2024, 440 pages (SRAD_Ex- B-02_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-03, Dec. 9, 2024, 569 pages (SRAD_Ex- B-03_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-04, Dec. 9, 2024, 305 pages (SRAD_Ex- B-04_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-05, Dec. 9, 2024, 369 pages (SRAD_Ex- B-05_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-06, Dec. 9, 2024, 332 pages (SRAD_Ex- B-06_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-07, Dec. 9, 2024, 305 pages (SRAD_Ex- B-07_12-9-24).

(56) **References Cited**

OTHER PUBLICATIONS

Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-08, Dec. 9, 2024, 145 pages (SRAD_Ex- B-08_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-09, Dec. 9, 2024, 108 pages (SRAD_Ex- B-09_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-10, Dec. 9, 2024, 125 pages (SRAD_Ex- B-10_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-11, Dec. 9, 2024, 189 pages (SRAD_Ex- B-11_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-12, Dec. 9, 2024, 308 pages (SRAD_Ex- B-12_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-13, Dec. 9, 2024, 281 pages (SRAD_Ex- B-13_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-14, Dec. 9, 2024, 412 pages (SRAD_Ex- B-14_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-15, Dec. 9, 2024, 195 pages (SRAD_Ex- B-15_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-16, Dec. 9, 2024, 976 pages (SRAD_Ex- B-16_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-17, Dec. 9, 2024, 590 pages (SRAD_Ex- B-17_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-18, Dec. 9, 2024, 363 pages (SRAD_Ex- B-18_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-20, Dec. 9, 2024, 255 pages (SRAD_Ex- B-20_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-21, Dec. 9, 2024, 430 pages (SRAD_Ex- B-21_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-22, Dec. 9, 2024, 682 pages (SRAD_Ex- B-22_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-23, Dec. 9, 2024, 323 pages (SRAD_Ex- B-23_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-24, Dec. 9, 2024, 488 pages (SRAD_Ex- B-24_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-25, Dec. 9, 2024, 573 pages (SRAD_Ex- B-25_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-26, Dec. 9, 2024, 256 pages (SRAD_Ex- B-26_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-27, Dec. 9, 2024, 171 pages (SRAD_Ex- B-27_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-28, Dec. 9, 2024, 267 pages (SRAD_Ex- B-28_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-30, Dec. 9, 2024, 647 pages (SRAD_Ex- B-30_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-31, Dec. 9, 2024, 439 pages (SRAD_Ex- B-31_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-32, Dec. 9, 2024, 278 pages (SRAD_Ex- B-32_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-33, Dec. 9, 2024, 323 pages (SRAD_Ex- B-33_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-34, Dec. 9, 2024, 385 pages (SRAD_Ex- B-34_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-35, Dec. 9, 2024, 375 pages (SRAD_Ex- B-35_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-36, Dec. 9, 2024, 116 pages (SRAD_Ex- B-36_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-37, Dec. 9, 2024, 398 pages (SRAD_Ex- B-37_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-38, Dec. 9, 2024, 15 pages (SRAD_Ex- B-38_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-39, Dec. 9, 2024, 30 pages (SRAD_Ex- B-39_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex B-40, Dec. 9, 2024, 29 pages (SRAD_Ex- B-40_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-01, Dec. 9, 2024, 257 pages (SRAD_Ex- C-01_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-02, Dec. 9, 2024, 594 pages (SRAD_Ex- C-02_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-03, Dec. 9, 2024, 636 pages (SRAD_Ex- C-03_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-04, Dec. 9, 2024, 295 pages (SRAD_Ex- C-04_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-05, Dec. 9, 2024, 279 pages (SRAD_Ex- C-05_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-06, Dec. 9, 2024, 373 pages (SRAD_Ex- C-06_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-23, Dec. 9, 2024, 405 pages (SRAD_Ex- D-23_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-24, Dec. 9, 2024, 325 pages (SRAD_Ex- D-24_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-25, Dec. 9, 2024, 571 pages (SRAD_Ex- D-25_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-26, Dec. 9, 2024, 127 pages (SRAD_Ex- D-26_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-27, Dec. 9, 2024, 146 pages (SRAD_Ex- D-27_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-28, Dec. 9, 2024, 139 pages (SRAD_Ex- D-28_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-30, Dec. 9, 2024, 515 pages (SRAD_Ex- D-30_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-31, Dec. 9, 2024, 264 pages (SRAD_Ex- D-31_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-32, Dec. 9, 2024, 268 pages (SRAD_Ex- D-32_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-33, Dec. 9, 2024, 191 pages (SRAD_Ex- D-33_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-34, Dec. 9, 2024, 211 pages (SRAD_Ex- D-34_12-9-24).

(56) **References Cited**

OTHER PUBLICATIONS

Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-35, Dec. 9, 2024, 211 pages (SRAD_Ex- D-35_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-36, Dec. 9, 2024, 87 pages (SRAD_Ex- D-36_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-37, Dec. 9, 2024, 252 pages (SRAD_Ex- D-37_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-40, Dec. 9, 2024, 11 pages (SRAD_Ex- D-40_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex E-01, Dec. 9, 2024, 33 pages (SRAD_Ex- E-01_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex E-02, Dec. 9, 2024, 31 pages (SRAD_Ex- E-02_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex E-03, Dec. 9, 2024, 30 pages (SRAD_Ex- E-03_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex E-04, Dec. 9, 2024, 28 pages (SRAD_Ex- E-04_12-9-24).
Sportscastr Inc. (Patent Owner) Preliminary Response In IPR-2024-01307 served Dec. 19, 2024, 81 pages (IPR-2024-01307_Sportcastr_Prelim_Res_12-19-24).
Sportscastr Inc. (Patent Owner) Preliminary Response In IPR-2024-01310 served Dec. 20, 2024, 84 pages (IPR-2024-01310_Sportcastr_Prelim_Res_12-20-24).
Sportscastr Inc. (Patent Owner) Preliminary Response In IPR-2024-01311 served Dec. 20, 2024, 87 pages (IPR-2024-01311_Sportcastr_Prelim_Resp_12-20-24).
Sportscastr Inc. (Patent Owner) Preliminary Response, IPR2024-01305, served Dec. 19, 2024, 81 pages (IPR-2024-01305_Sportcastr_Prelim_Res_12-19-24).
U. Rauschenbach, "A scalable interactive TV service supporting synchronized delivery over broadcast and broadband networks." IBC 2004 Conference, (Sep. 2004), https://www.rauschenbach.net/Publications/docs/paper-IBC04-SAVANT-rauschenbach_et_al.pdf, 8 pages (Rauschenbach_et_al_Scalable_IBC_2004).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-07, Dec. 9, 2024, 309 pages (SRAD_Ex- C-07_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-08, Dec. 9, 2024, 330 pages (SRAD_Ex- C-08_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-09, Dec. 9, 2024, 243 pages (SRAD_Ex- C-09_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-10, Dec. 9, 2024, 327 pages (SRAD_Ex- C-10_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-11, Dec. 9, 2024, 233 pages (SRAD_Ex- C-11_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-12, Dec. 9, 2024, 322 pages (SRAD_Ex- C-12_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-13, Dec. 9, 2024, 580 pages (SRAD_Ex- C-13_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-14, Dec. 9, 2024, 976 pages (SRAD_Ex- C-14_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-15, Dec. 9, 2024, 441 pages (SRAD_Ex- C-15_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-16, Dec. 9, 2024, 1093 pages (SRAD_Ex- C-16_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-17, Dec. 9, 2024, 449 pages (SRAD_Ex- C-17_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-18, Dec. 9, 2024, 365 pages (SRAD_Ex- C-18_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-20, Dec. 9, 2024, 388 pages (SRAD_Ex- C-20_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-21, Dec. 9, 2024, 694 pages (SRAD_Ex- C-21_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-22, Dec. 9, 2024, 525 pages (SRAD_Ex- C-22_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-23, Dec. 9, 2024, 662 pages (SRAD_Ex- C-23_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-24, Dec. 9, 2024, 480 pages (SRAD_Ex- C-24_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-25, Dec. 9, 2024, 761 pages (SRAD_Ex- C-25_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-26, Dec. 9, 2024, 250 pages (SRAD_Ex- C-26_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-27, Dec. 9, 2024, 199 pages (SRAD_Ex- C-27_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-28, Dec. 9, 2024, 315 pages (SRAD_Ex- C-28_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-30, Dec. 9, 2024, 878 pages (SRAD_Ex- C-30_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-31, Dec. 9, 2024, 369 pages (SRAD_Ex- C-31_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-32, Dec. 9, 2024, 372 pages (SRAD_Ex- C-32_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-33, Dec. 9, 2024, 270 pages (SRAD_Ex- C-33_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-34, Dec. 9, 2024, 335 pages (SRAD_Ex- C-34_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-35, Dec. 9, 2024, 342 pages (SRAD_Ex- C-35_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-36, Dec. 9, 2024, 122 pages (SRAD_Ex- C-36_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex C-37, Dec. 9, 2024, 363 pages (SRAD_Ex- C-37_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-01, Dec. 9, 2024, 150 pages (SRAD_Ex- D-01_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-02, Dec. 9, 2024, 328 pages (SRAD_Ex- D-02_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-03, Dec. 9, 2024, 264 pages (SRAD_Ex- D-03_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-04, Dec. 9, 2024, 195 pages (SRAD_Ex- D-04_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-05, Dec. 9, 2024, 173 pages (SRAD_Ex- D-05_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-06, Dec. 9, 2024, 200 pages (SRAD_Ex- D-06_12-9-24).

(56) **References Cited**

OTHER PUBLICATIONS

Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-07, Dec. 9, 2024, 187 pages (SRAD_Ex- D-07_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-08, Dec. 9, 2024, 182 pages (SRAD_Ex- D-08_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-09, Dec. 9, 2024, 146 pages (SRAD_Ex- D-09_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-10, Dec. 9, 2024, 193 pages (SRAD_Ex- D-10_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-11, Dec. 9, 2024, 118 pages (SRAD_Ex- D-11_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-12, Dec. 9, 2024, 180 pages (SRAD_Ex- D-12_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-13, Dec. 9, 2024, 263 pages (SRAD_Ex- D-13_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-14, Dec. 9, 2024, 387 pages (SRAD_Ex- D-14_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-15, Dec. 9, 2024, 221 pages (SRAD_Ex- D-15_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-16, Dec. 9, 2024, 665 pages (SRAD_Ex- D-16_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-17, Dec. 9, 2024, 321 pages (SRAD_Ex- D-17_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-18, Dec. 9, 2024, 232 pages (SRAD_Ex- D-18_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-20, Dec. 9, 2024, 235 pages (SRAD_Ex- D-20_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-21, Dec. 9, 2024, 503 pages (SRAD_Ex- D-21_12-9-24).
Sportradar and Sportradar Group AG Amended Invalidity Contentions, Ex D-22, Dec. 9, 2024, 359 pages (SRAD_Ex- D-22_12-9-24).
Comparison of the KIA Declaration dated Dec. 6, 2024 to WICKER Declaration dated Aug. 22, 2024, 508 pages (Comparison_KIA_WICKER).
Curriculum Vitae of Dr. Omid Kia, Ph.D., 13 pages (KIA_CV).
Declaration of Dr. Omid Kia, Ph.D., for Patents 10,425,697 and 10,805,687, dated Dec. 13, 2024, 392 pages (Kia_Dec_re_10425697_and_10805687_12-13-24_392_pg).
Declaration of Dr. Omid Kia, Ph.D.for IPR Nos. 2025-00265, 2025-00266, 2025-00268, 2025-00269, 2025-00273, 2025-00275 dated Dec. 6, 2024, 502 pages (Kia_Decl_12-6-24).
Gilstrap Order of Sep. 30, 2024 authorizing addition of Challenged Claims to the Complaint, 2 pages (Gilstrap_Order_09-30-2024).
Pantos et al., "HTTP Live Streaming," ("Pantos") published Apr. 4, 2016, 33 pages (PANTOS_04-04-2016_33_pg).
Pantos, Wayback Machine Archive dated Apr. 4, 2016, 1 page (PANTOS_Wayback_Machine_04-Apr. 2026).
Petition for Inter Partes Review of U.S. Pat. No. 10,425,697 (IPR 2025-00265), filed Dec. 11, 2024, 98 pages (IPR2025-00265_Petition_12-11-2024).
Petition for Inter Partes Review of U.S. Pat. No. 10,425,697 (IPR 2025-00266), filed Dec. 11, 2024, 101 pages (IPR2025-00266_Petition_12-11-2024).
Petition for Inter Partes Review of U.S. Pat. No. 10,425,697 (IPR 2025-00313), filed Dec. 14, 2024, 95 pages (IPR2024-00313_Petition_12-14-2024).
Petition for Inter Partes Review of U.S. Pat. No. 10,425,697 (IPR 2025-00314), filed Dec. 14, 2024, 90 pages (IPR2024-00314_Petition_12-14-2024).
Petition for Inter Partes Review of U.S. Pat. No. 10,805,687 (IPR 2025-00268), filed Dec. 11, 2024, 100 pages (IPR2025-00268_Petition_12-11-2024).
Petition for Inter Partes Review of U.S. Pat. No. 10,805,687 (IPR 2025-00269), filed Dec. 11, 2024, 102 pages (IPR2025-00269_Petition_12-11-2024).
Petition for Inter Partes Review of U.S. Pat. No. 10,805,687 (IPR 2025-00315), filed Dec. 14, 2024, 95 pages (IPR2024-00315_Petition_12-14-2024).
Petition for Inter Partes Review of U.S. Pat. No. 10,805,687 (IPR 2025-00316), filed Dec. 14, 2024, 86 pages (IPR2024-00316_Petition_12-14-2024).
Petition for Inter Partes Review of U.S. Pat. No. 11,039,218 (IPR 2025-00273), filed Dec. 11, 2024, 101 pages (IPR2025-00273_Petition_12-11-2024).
Petition for Inter Partes Review of U.S. Pat. No. 11,039,218 (IPR 2025-00275), filed Dec. 11, 2024, 102 pages (IPR2025-00275_Petition_12-11-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 10,425,697 (IPR 2025-00265), filed Dec. 11, 2024, 7 pages (IPR2025-00265_Petitioner_Showing_Cause_12-11-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 10,425,697 (IPR 2025-00266), filed Dec. 11, 2024, 7 pages(IPR2025-00266_Petitioner_Showing_Cause_12-11-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 10,425,697 (IPR 2025-00313), filed Dec. 14, 2024, 6 pages (IPR2024-00313_Petitioner_Showing_Cause_12-14-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 10,425,697 (IPR 2025-00314), filed Dec. 14, 2024, 6 pages (IPR2024-00314_Petitioner_Showing_Cause_12-14-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 10,805,687 (IPR 2025-00268), filed Dec. 11, 2024, 7 pages(IPR2025-00268_Petitioner_Showing_Cause_12-11-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 10,805,687 (IPR 2025-00269), filed Dec. 11, 2024, 7 pages(IPR2025-00269_Petitioner_Showing_Cause_12-11-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 10,805,687 (IPR 2025-00315), filed Dec. 14, 2024, 6 pages (IPR2024-00315_Petitioner_Showing_Cause_12-14-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 10,805,687 (IPR 2025-00316), filed Dec. 14, 2024, 6 pages (IPR2024-00316_Petitioner_Showing_Cause_12-14-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 11,039,218 (IPR 2025-00273), filed Dec. 11, 2024, 6 pages(IPR2025-00273_Petitioner_Showing_Cause_12-11-2024).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 11,039,218 (IPR 2025-00275), filed Dec. 11, 2024, 6 pages(IPR2025-00275_Petitioner_Showing_Cause_12-11-2024).
SportsCastr's Disclosure of Asserted Claims and Preliminary Infringement Contentions, filed Jan. 18, 2024, 638 pages (SPORTSCASTR_Infring_Con_01-18-2024).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-13S, Dec. 9, 2024, 21 pages (GEN-Ex-A-13S_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-14S. Dec. 9, 2024, 21 pages (GEN-Ex-A-14S_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-15, Dec. 9, 2024, 57 pages (GEN-Ex-A-15_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-16, Dec. 9, 2024, 61 pages (GEN-Ex-A-16_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-18, Dec. 9, 2024, 58 pages (GEN-Ex-A-18_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-20, Dec. 9, 2024, 54 pages (GEN-Ex-A-20_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-21, Dec. 9, 2024, 55 pages (GEN-Ex-A-21_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-22, Dec. 9, 2024, 52 pages (GEN-Ex-A-22_12-9-24).

(56) **References Cited**

OTHER PUBLICATIONS

Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-23, Dec. 9, 2024, 54 pages (GEN-Ex-A-23_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-24, Dec. 9, 2024, 53 pages (GEN-Ex-A-24_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex A-25, Dec. 9, 2024, 63 pages (GEN-Ex-A-25_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-26, Dec. 9, 2024, 57 pages (GEN-Ex-A-26_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-27, Dec. 9, 2024, 59 pages (GEN-Ex-A-27_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-28, Dec. 9, 2024, 58 pages (GEN-Ex-A-28_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-29, Dec. 9, 2024, 55 pages (GEN-Ex-A-29_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-30, Dec. 9, 2024, 68 pages (GEN-Ex-A-30_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-31, Dec. 9, 2024, 55 pages (GEN-Ex-A-31_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-32, Dec. 9, 2024, 53 pages (GEN-Ex-A-32_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-33, Dec. 9, 2024, 59 pages (GEN-Ex-A-33_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-34, Dec. 9, 2024, 54 pages (GEN-Ex-A-34_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex A-35, Dec. 9, 2024, 55 pages (GEN-Ex-A-35_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-01S, Dec. 9, 2024, 127 pages (GEN-Ex-B-01S_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-02S, Dec. 9, 2024, 183 pages (GEN-Ex-B-02S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-03S, Dec. 9, 2024, 164 pages (GEN-Ex-B-03S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-04S, Dec. 9, 2024, 276 pages (GEN-Ex-B-04S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-05S, Dec. 9, 2024, 219 pages (GEN-Ex-B-05S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-06S, Dec. 9, 2024, 137 pages (GEN-Ex-B-06S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-07S, Dec. 9, 2024, 145 pages (GEN-Ex-B-07S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-08S, Dec. 9, 2024, 74 pages (GEN-Ex-B-08S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-09S, Dec. 9, 2024, 107 pages (GEN-Ex-B-09S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-10S, Dec. 9, 2024, 281 pages (GEN-Ex-B-10S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-11S, Dec. 9, 2024, 105 pages (GEN-Ex-B-11S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-12S, Dec. 9, 2024, 344 pages (GEN-Ex-B-12S_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-13S, Dec. 9, 2024, 37 pages (GEN-Ex-B-13S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-14S, Dec. 9, 2024, 110 pages (GEN-Ex-B-14S_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-15, Dec. 9, 2024, 122 pages (GEN-Ex-B-15_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-16, Dec. 9, 2024, 224 pages (GEN-Ex-B-16_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-18, Dec. 9, 2024, 203 pages (GEN-Ex-B-18_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-20, Dec. 9, 2024, 84 pages (GEN-Ex-B-20_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-21, Dec. 9, 2024, 82 pages (GEN-Ex-B-21_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-22, Dec. 9, 2024, 84 pages (GEN-Ex-B-22_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-23, Dec. 9, 2024, 81pages (GEN-Ex-B-23_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-24, Dec. 9, 2024, 82 pages (GEN-Ex-B-24_12-9-24).
Sports' 2nd Supp. Invalidity Contentions, Ex B-25, Dec. 9, 2024, 189 pages (GEN-Ex-B-25_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-26, Dec. 9, 2024, 78 pages (GEN-Ex-B-26_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-27, Dec. 9, 2024, 527 pages (GEN-Ex-B-27_12-9-24) (10 parts.
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-28, Dec. 9, 2024, 133 pages (GEN-Ex-B-28_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-29, Dec. 9, 2024, 363 pages (GEN-Ex-B-29_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-30, Dec. 9, 2024, 315 pages (GEN-Ex-B-30_12-9-24).
Genius Sports' 2nd Supp. Invalidity Contentions, Ex B-31, Dec. 9, 2024, 279 pages (GEN-Ex-B-31_12-9-24).
Comparison of the Declaration of Dr. Omid Kia, PhD. (Feb. 11, 2025) with Wicker Declaration (Dec. 6, 2024) 232 pages (Comparison_Kia_Wicker_232-pages), 4 parts.
Declaration of D. Omid Kia, Ph.D., Feb. 11, 2025, 224 pages (Kia_Decl_02-11-2025_224page), 4 parts.
Infrared5's Red5 Pro Streaming Application and website documents, cited in Exhibit A-34 of Genius Contentions (identified in Exhibit as GS_00022230-22290) dated Dec. 19, 2024, 61 pages (Red5-Pro).
Nerdordie, Going live on Twitch with OBS Studio, YOUTUBE (Apr. 15, 2016) https://www.youtube? v=aOHhxOolre&list=PLT3Ure7_kYHwj80T3AV-pZ4_r7yp6mDg-&index=13 (GS_00022420_w_transcription), document provided natively, transcript provided, 4 pages.
Nerdordie, Scenes in OBS Studio, YOUTUBE (Apr. 13, 2015) https://www.youtube.com/watch? v=hbjQIxyLBCE&list=PLT3Ure7_kYHwj80T3AV-pZ4_r7yp6mDg-&index+9 (GS_00022421_w_transcription), document provided natively, transcript provided, 14 pages.
Nerdordie, Setting Up Twitch Account and Nightbot Before We Streem, YOUTUBE (Apr. 15, 2015) https://www. youtube.com/watch?v=789sp75dYyk&list=PLT3Ure7_kYHwj80T2AV-pZ4_r7yp6mDg-&index=12 (GS_00022419_w_transcription), document provided natively, transcript provided, 12 pages.
Nerdordie, what Twitch Alert system Should You Use? YOUTUBE (Apr. 13, 2016) https://www.youtube.com/watch? v=SJt0NW41dOc&list=PLT3Ure7_kYHwj80T3AV-pZ4_r7yp6mDg-&index=11 (GS_00022423_w_transcription) document provided natively, transcript provided, 11 pages.
Petition for Inter Partes Review of U.S. Pat. No. 11,871,088 (IPR 2025-00634), filed Feb. 24, 2025, (IPR2025-00634_Petition_2-24-2025) 102 pages.
Petition for Inter Partes Review of U.S. Pat. No. 11,871,088 (IPR 2025-00635), filed Feb. 24, 2025, (IPR2025-00635_Petition_2-24-2025) 100 pages.
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 11,871,088 (IPR 2025-00634), filed Feb. 24, 2025, 6 pages (IPR2025-00634_Petitioner_Showing_Cause_2-24-2025).
Petitioner Showing of Good Cause for Parallel Petitions of U.S. Pat. No. 11,871,088 (IPR 2025-00635), filed Feb. 24, 2025, 6 pages (IPR2025-00635_Petitioner_Showing_Cause_2-24-2025).
Ptzoptics, Live Streaming Video Game Contest to Twitch—League of Legends, YOUTUBE (Dec. 16, 2015) https://www.youtube.com/watch?v=9bGADGoK3is (GS_00022425_w_transcription) document provided natively, transcript provided, 7 page.
Publish-Subscribe Systems, CloudAMQP, cited in Exhibit A-35 of Genius Contentions (identified in Exhibit as GS_00021220-21247) dated Dec. 19, 2024, 28 pages (CloudAMQP).
Publish-Subscribe Systems, PubNub, cited in Exhibit A-35 of Genius Contentions (identified in Exhibit as GS_00021272-GS_21539) dated Dec. 19, 2024, 261 pages (PubNub), 3 parts.
Publish-Subscribe Systems, Pusher, cited in Exhibit A-35 of Genius Contentions (identified in Exhibit as GS_00021540-GS_21722) dated Dec. 19, 2024, 183 pages (Pusher), 2 parts.
Publish-Subscribe Systems, RabbitMQ, cited in Exhibit A-35 of Genius Contentions (identified in Exhibit as GS_00021723-22229) dated Dec. 19, 2024, 262 pages (RabbitMQ), 4 parts.

(56) **References Cited**

OTHER PUBLICATIONS

Publish-Subscribe Systems, Redis, cited in Exhibit A-35 of Genius Contentions (identified in Exhibit as GS_00022291-22314) dated Dec. 19, 2024, 23 pages (Redis).

Publish-Subscribe Systems, Socket.IO, cited in Exhibit A-35 of Genius Contentions (identified in Exhibit as GS_00022315-22387) dated Dec. 19, 2024, 73 pages (SocketIO).

Publish-Subscribe Systems, ZeroMQ, cited in Exhibit A-35 of Genius Contentions (identified in Exhibit as GS_00022612-24285) dated Dec. 19, 2024, 502 pages (ZeroMQ), 9 parts.

Stuart Dredge, What is Twitch, and why does Google want it? The Guardian (May 19, 2014) https://www.theguardian. com/technology/2014/Maymay/19/twitch-youtube-live-games-google-acquisition-pokemon (Stuart_Dredge_GS_0022430), 3 pages.

Twitch 2015 Retrospective, https://webarchive.org/web/20160607170832/htt;://twitch.tv/year/2015, 28 pages (twitch-2015_GS 00022391).

Twitch Announces Open Developer Program with New Tools to Enable Gameplay Highlight, Discover, Creation, and Sharing. BUSINESSWIRE (Jun. 10, 2013) https//www.businesswire.com/news/home/201306100005347/en/Twitch- Announces-Open-Developer-Program-With-New-Tools-To-Enable-Gameplay-Highlight-Discovery-Creation-and-Sharing (Twitch_Announces_GS_00022453), 3 pages.

Twitch on the App Store, https://web.archive.org/web/20160530200407/htt;//itunes.apple.com/US/app/twitchtv/ id460177396, (Twitch_app_GS_00022535), 2 pages.

Twitch's API documentation, including TWITCH_0000001-000291, published online at https://github.com/justintv/Twitch-API, (Twitch_API_Documentation_291pages), 4 parts.

Twitch's customer help documentation, production Nos. GS_00022456-22534, published online https://help.twitch.tv/, (TWITCH_Customer_help_doc_79pages) Nos. GS-00022433-2245, published online at http://www.twitch.tv/, 23.

Twitch's public-facing website, production pages (Twitch_PUblic-facing_website_23p_npl).

* cited by examiner

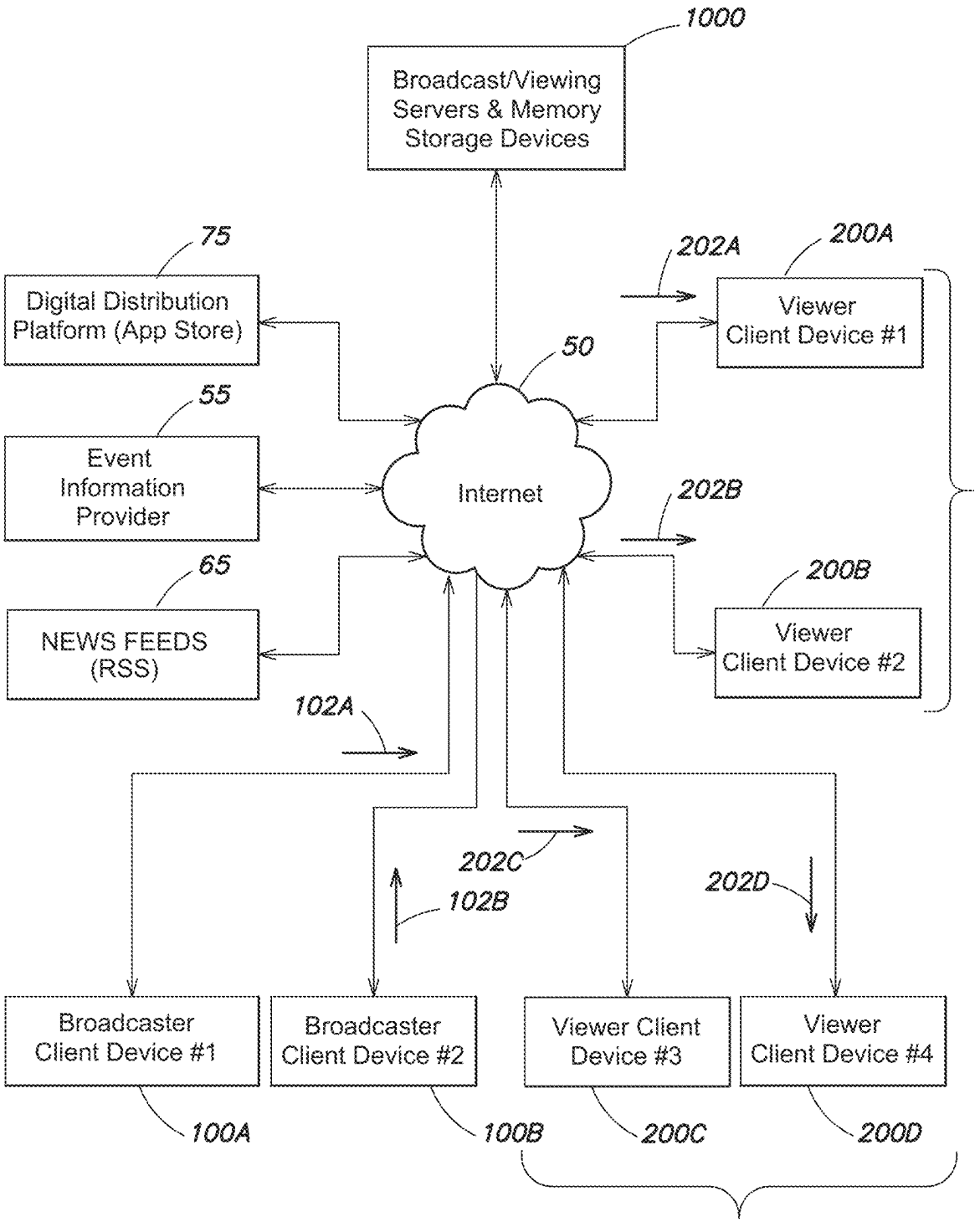

1000
Broadcast/Viewing Servers & Memory Storage Devices
75
Digital Distribution Platform (App Store)
55
Event Information Provider
65
NEWS FEEDS (RSS)
50
Internet
202A
200A
Viewer Client Device #1
202B
200B
Viewer Client Device #2
102A
202C
102B
202D
Broadcaster Client Device #1
Broadcaster Client Device #2
Viewer Client Device #3
Viewer Client Device #4
100A
100B
200C
200D

FIG. 1A

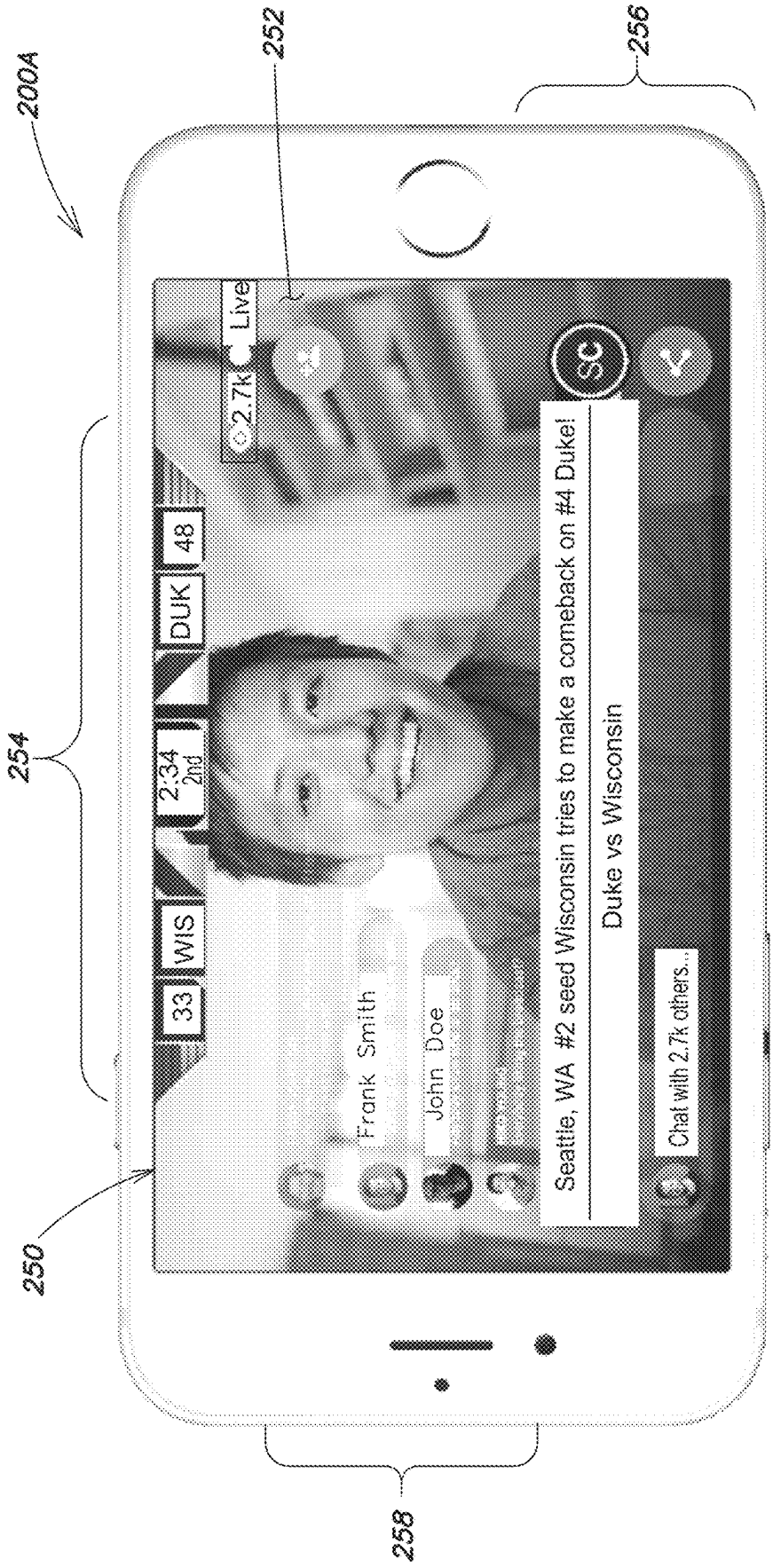

200A
252
256
254
250
258
33 WIS
2:34 2nd
DUK 48
2.7k Live
SC
Frank Smith
John Doe
Seattle, WA #2 seed Wisconsin tries to make a comeback on #4 Duke!
Duke vs Wisconsin
Chat with 2.7k others...

FIG. 1B

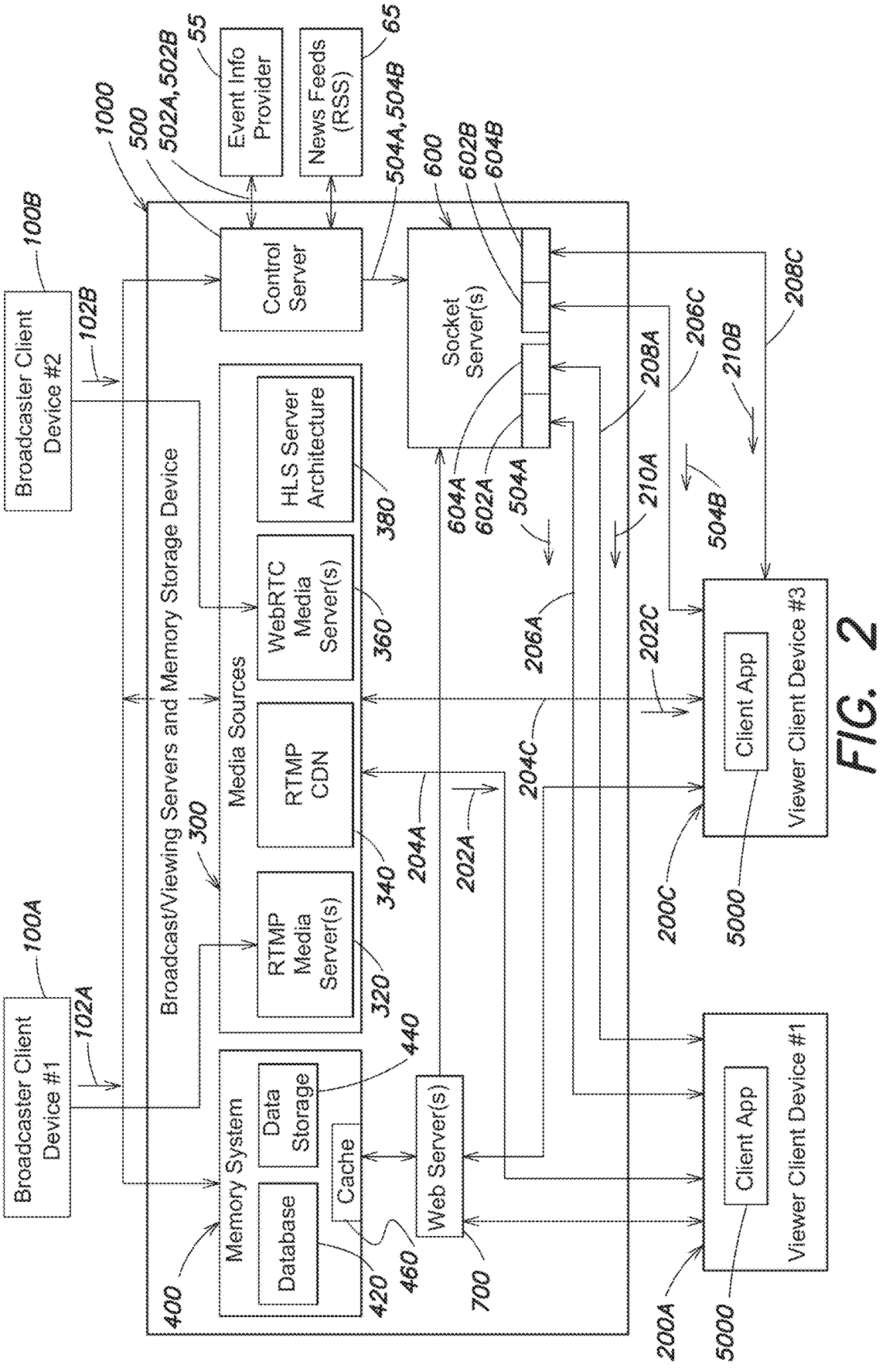

Broadcaster Client Device #1
100A
102A
Broadcaster Client Device #2
100B
102B
Broadcast/Viewing Servers and Memory Storage Device
300
1000
500
502A, 502B
55
Event Info Provider
News Feeds (RSS)
65
Control Server
504A, 504B
600
602B
604B
Socket Server(s)
604A
602A
504A
504B
208A
206C
210B
208C
210A
Media Sources
RTMP Media Server(s)
320
RTMP CDN
340
WebRTC Media Server(s)
360
HLS Server Architecture
380
206A
202C
204C
204A
202A
Memory System
Data Storage
440
Database
Cache
420
460
Web Server(s)
700
400
Client App
Viewer Client Device #3
200C
5000
Client App
Viewer Client Device #1
200A
5000

FIG. 2

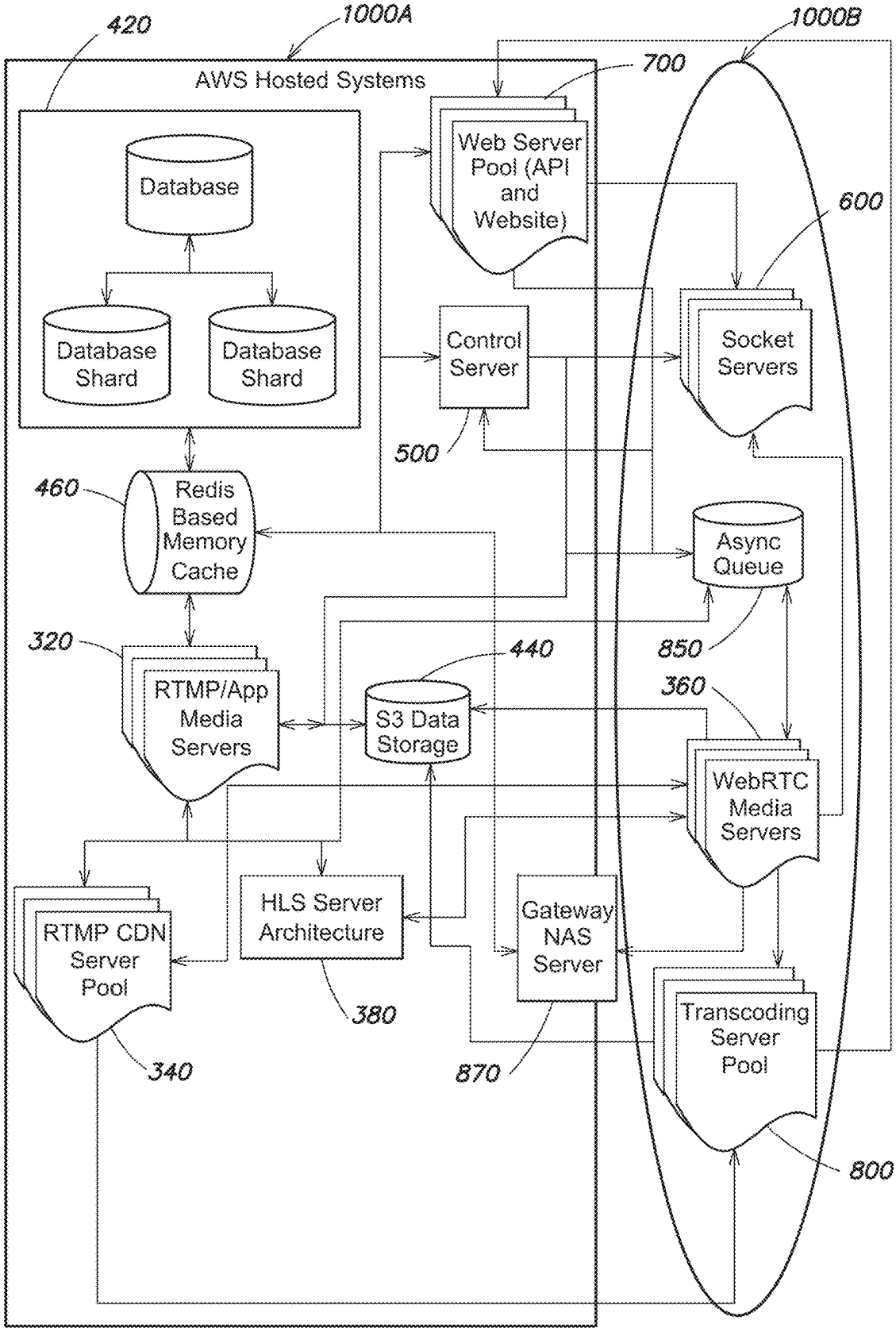

420
1000A
1000B
AWS Hosted Systems
700
Web Server Pool (API and Website)
Database
Database Shard
Database Shard
Control Server
500
600
Socket Servers
460
Redis Based Memory Cache
Async Queue
850
320
RTMP/App Media Servers
440
S3 Data Storage
360
WebRTC Media Servers
RTMP CDN Server Pool
340
HLS Server Architecture
380
Gateway NAS Server
870
Transcoding Server Pool
800

FIG. 3

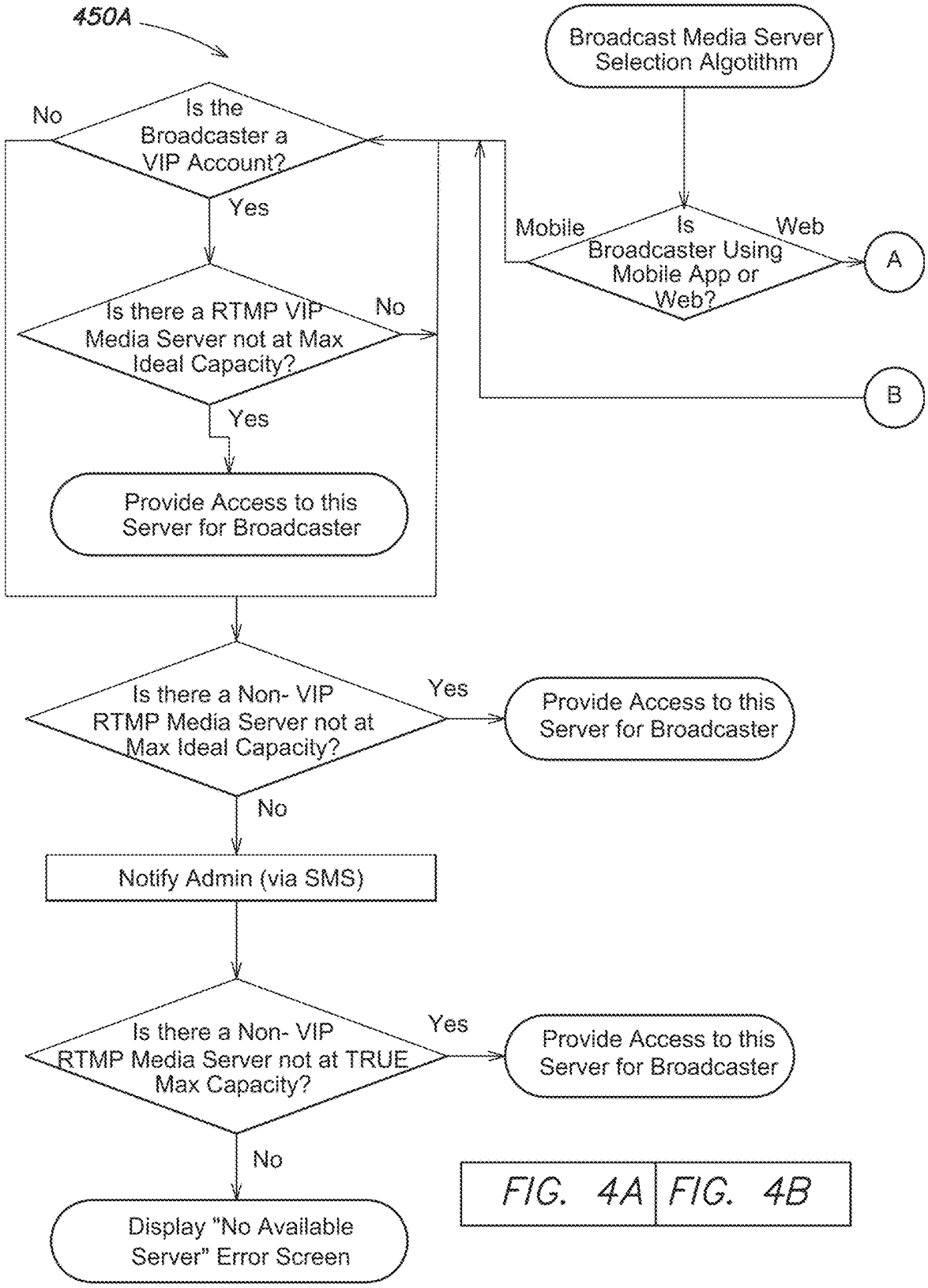


FIG. 4A | FIG. 4B

FIG. 4A

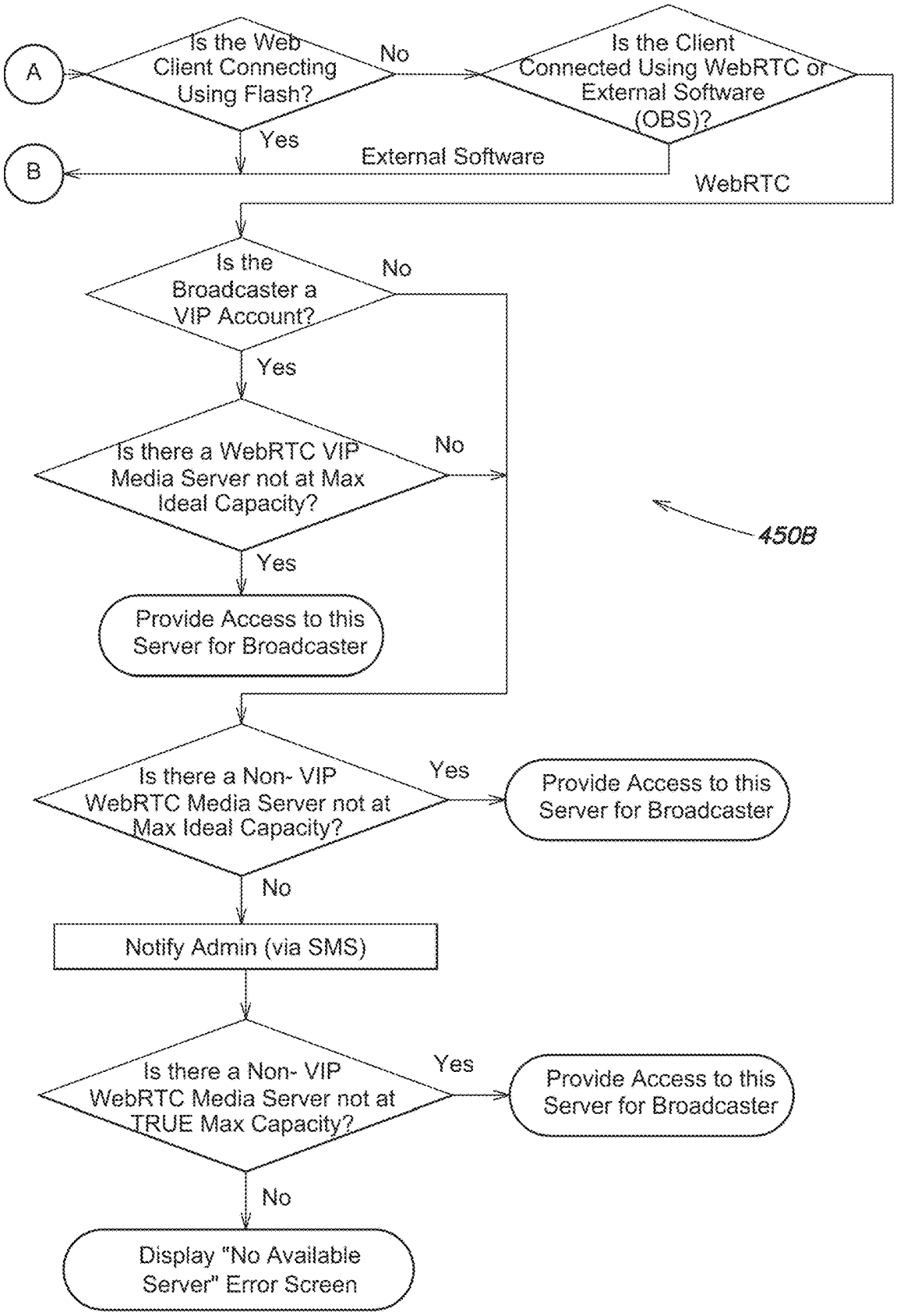

A
Is the Web Client Connecting Using Flash?
No
Is the Client Connected Using WebRTC or External Software (OBS)?
Yes
External Software
B
WebRTC
Is the Broadcaster a VIP Account?
No
Yes
Is there a WebRTC VIP Media Server not at Max Ideal Capacity?
No
Yes
Provide Access to this Server for Broadcaster
450B
Is there a Non- VIP WebRTC Media Server not at Max Ideal Capacity?
Yes
Provide Access to this Server for Broadcaster
No
Notify Admin (via SMS)
Is there a Non- VIP WebRTC Media Server not at TRUE Max Capacity?
Yes
Provide Access to this Server for Broadcaster
No
Display "No Available Server" Error Screen

FIG. 4B

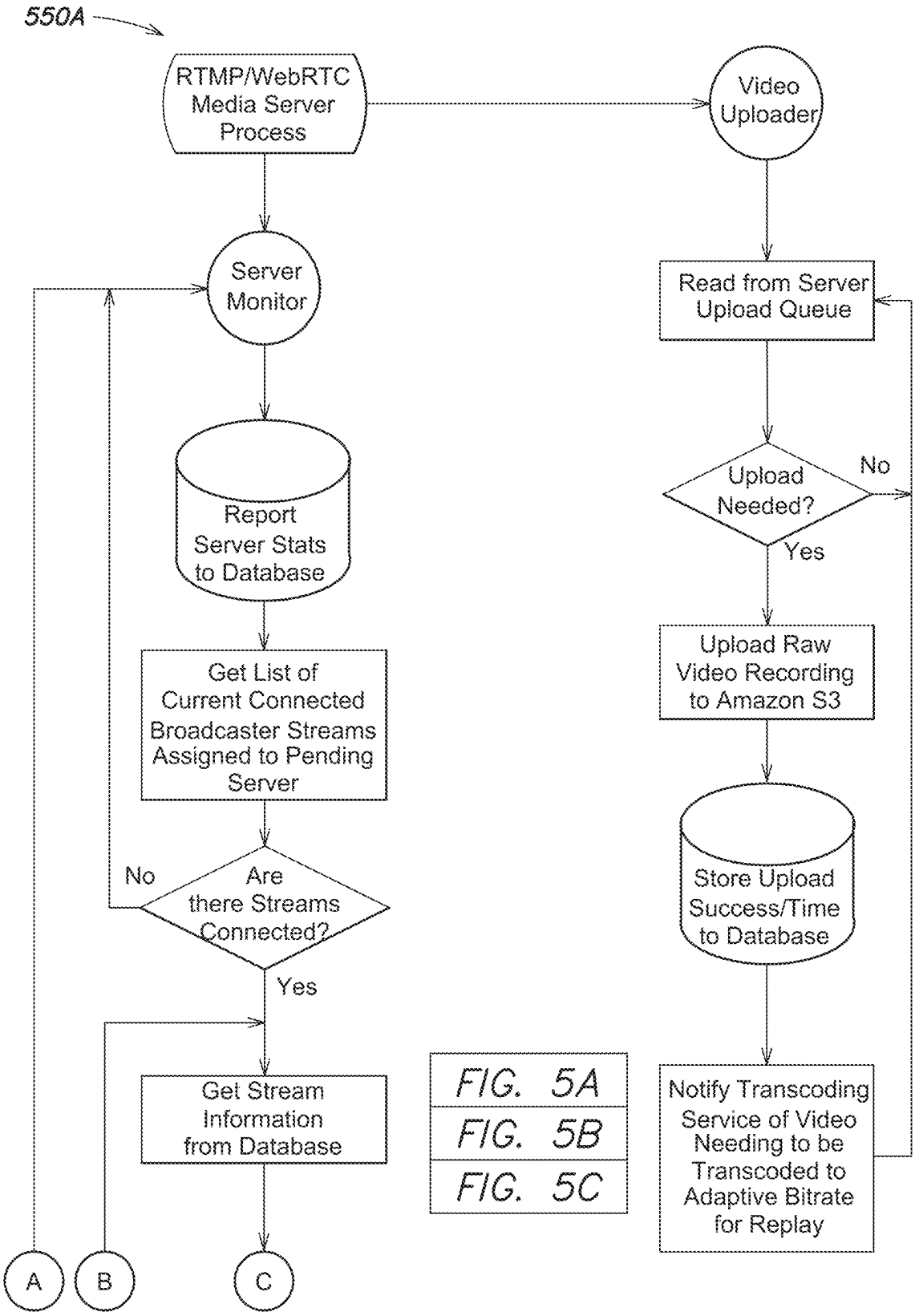


FIG. 5A
FIG. 5B
FIG. 5C

FIG. 5A

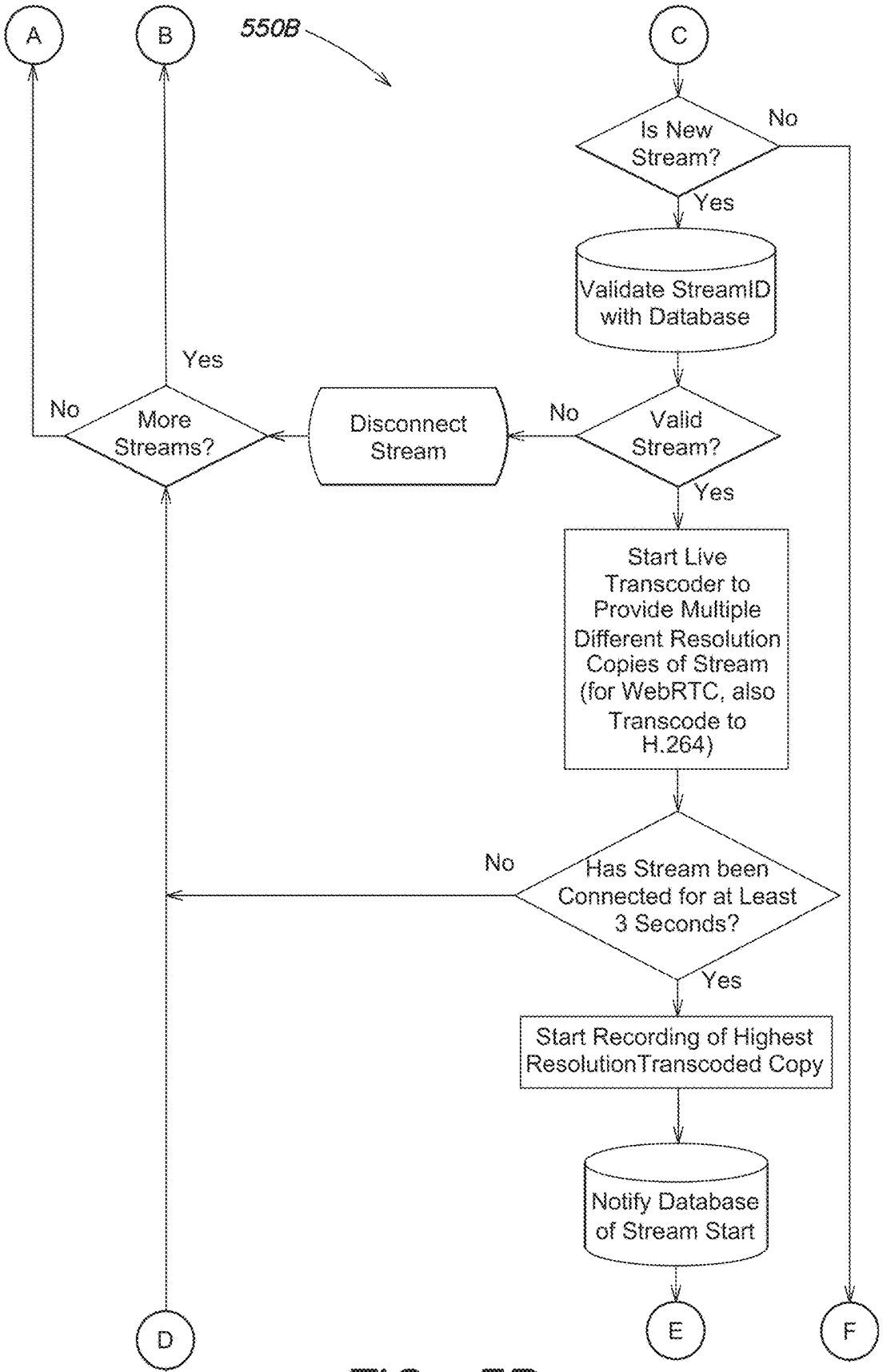

A
B
550B
C
Is New Stream?
No
Yes
Validate StreamID with Database
Yes
No
More Streams?
Disconnect Stream
No
Valid Stream?
Yes
Start Live Transcoder to Provide Multiple Different Resolution Copies of Stream (for WebRTC, also Transcode to H.264)
No
Has Stream been Connected for at Least 3 Seconds?
Yes
Start Recording of Highest ResolutionTranscoded Copy
Notify Database of Stream Start
D
E
F

FIG. 5B

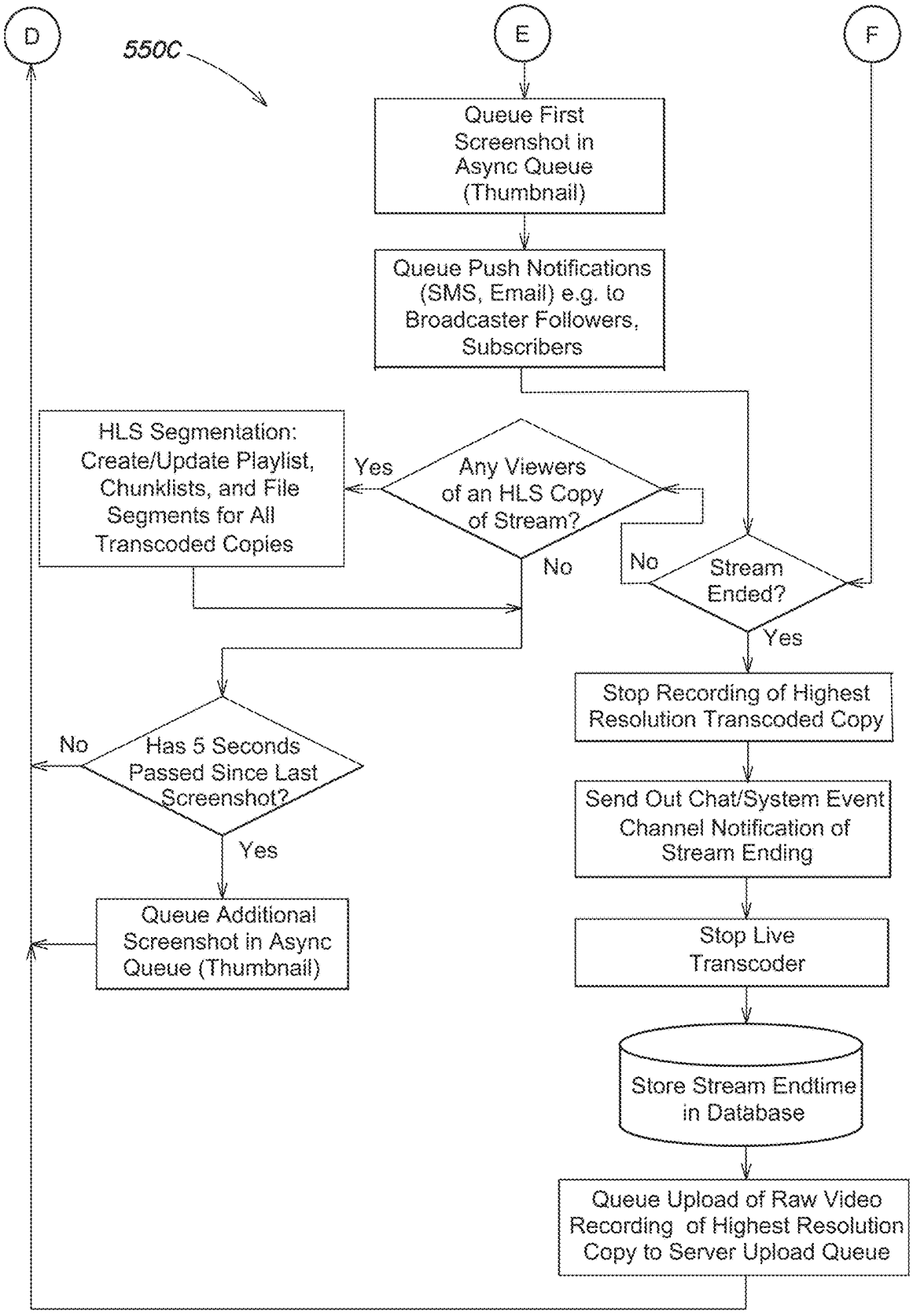

D
550C
E
F
Queue First Screenshot in Async Queue (Thumbnail)
Queue Push Notifications (SMS, Email) e.g. to Broadcaster Followers, Subscribers
HLS Segmentation: Create/Update Playlist, Chunklists, and File Segments for All Transcoded Copies
Yes
Any Viewers of an HLS Copy of Stream?
No
No
Stream Ended?
Yes
Stop Recording of Highest Resolution Transcoded Copy
No
Has 5 Seconds Passed Since Last Screenshot?
Yes
Send Out Chat/System Event Channel Notification of Stream Ending
Queue Additional Screenshot in Async Queue (Thumbnail)
Stop Live Transcoder
Store Stream Endtime in Database
Queue Upload of Raw Video Recording of Highest Resolution Copy to Server Upload Queue

FIG. 5C

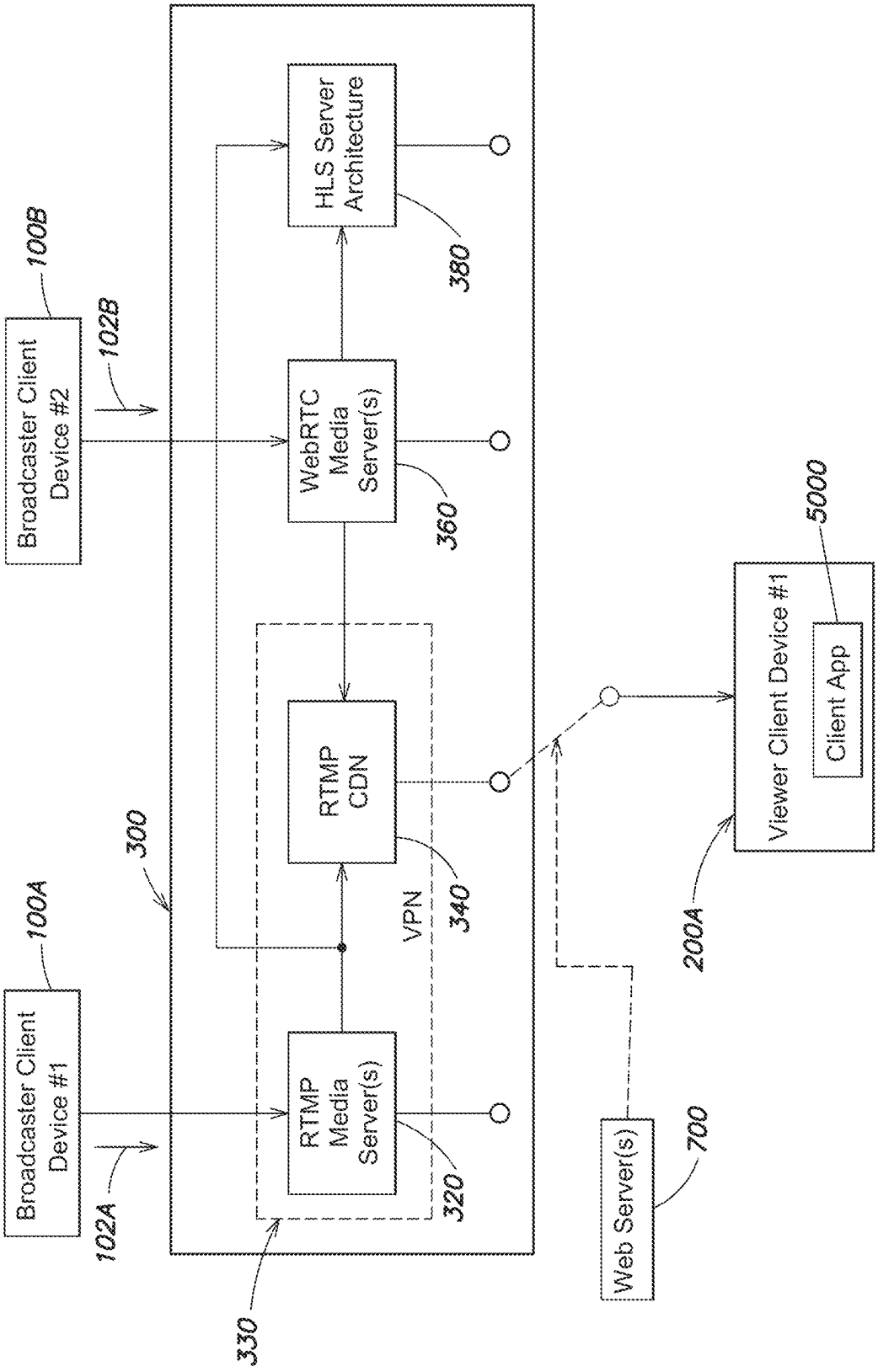

Broadcaster Client Device #1
100A
102A
Broadcaster Client Device #2
100B
102B
300
330
RTMP Media Server(s)
320
WebRTC Media Server(s)
360
HLS Server Architecture
380
RTMP CDN
340
VPN
Web Server(s)
700
Viewer Client Device #1
5000
Client App
200A

FIG. 6

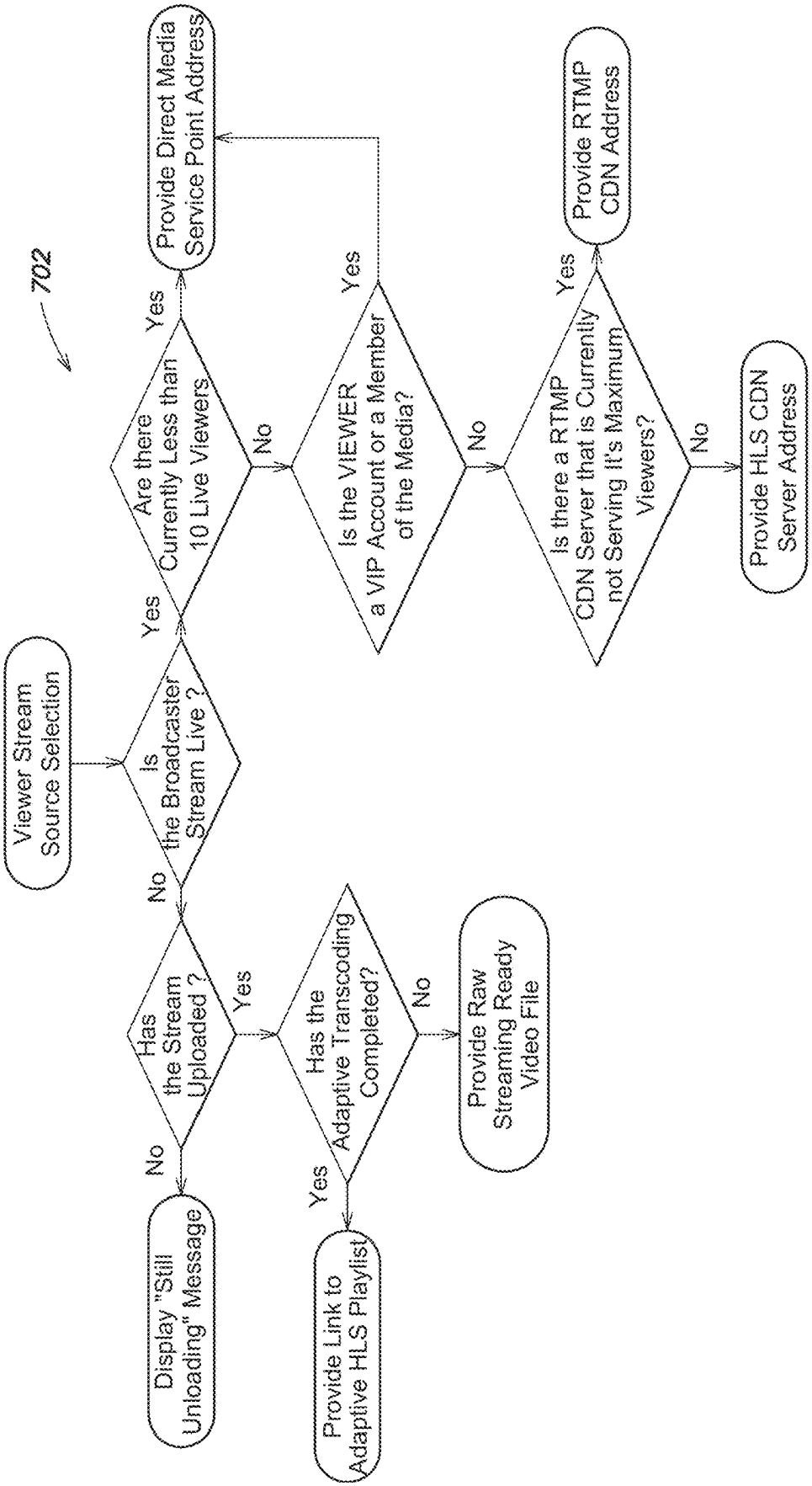

702
Viewer Stream Source Selection
Is the Broadcaster Stream Live ?
Yes
No
Are there Currently Less than 10 Live Viewers
Yes
No
Provide Direct Media Service Point Address
Is the VIEWER a VIP Account or a Member of the Media?
Yes
No
Is there a RTMP CDN Server that is Currently not Serving It's Maximum Viewers?
Yes
No
Provide RTMP CDN Address
Provide HLS CDN Server Address
Has the Stream Uploaded ?
No
Yes
Display "Still Unloading" Message
Has the Adaptive Transcoding Completed?
Yes
No
Provide Link to Adaptive HLS Playlist
Provide Raw Streaming Ready Video File

FIG. 7

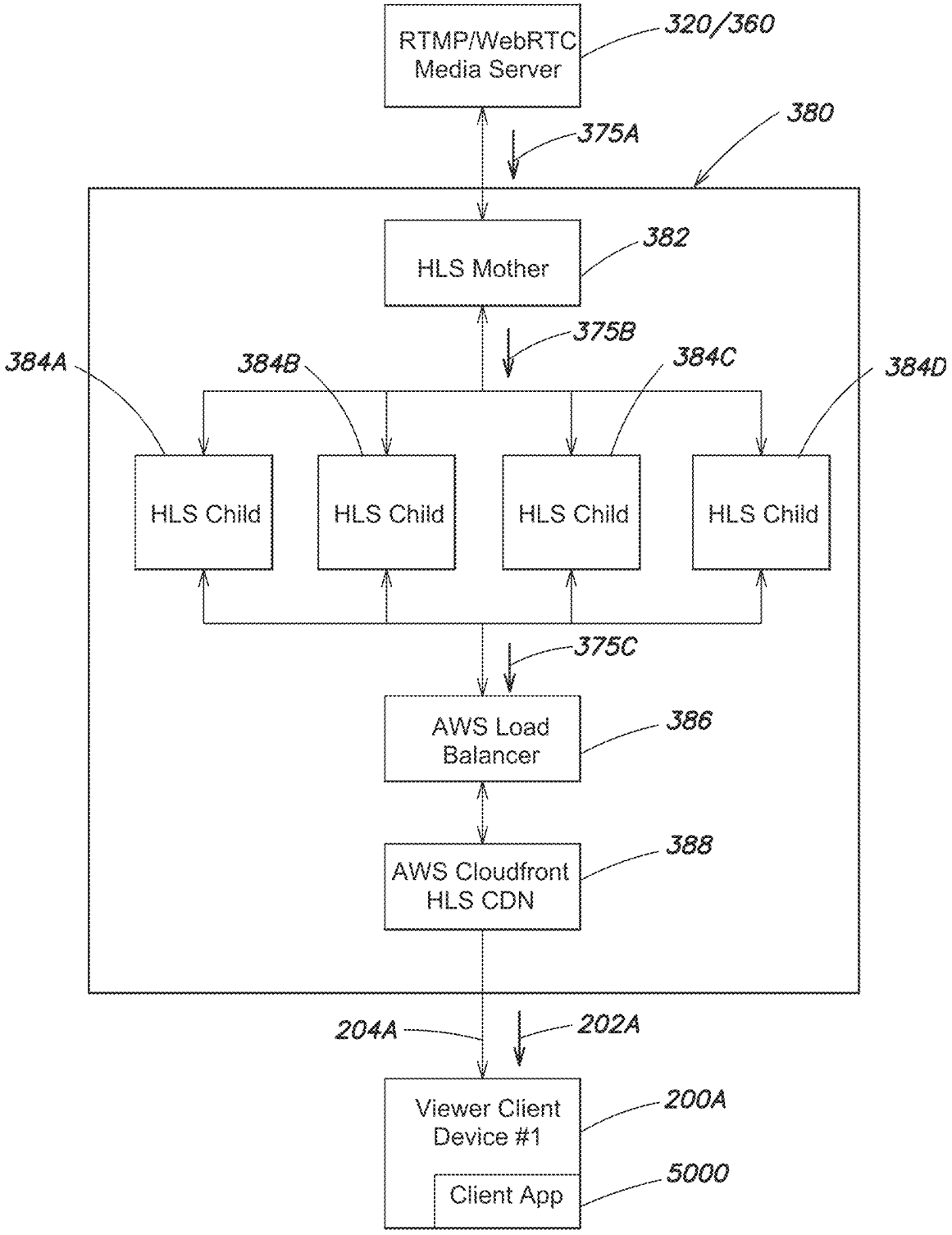

RTMP/WebRTC Media Server
320/360
380
375A
HLS Mother
382
375B
384A
384B
384C
384D
HLS Child
HLS Child
HLS Child
HLS Child
375C
AWS Load Balancer
386
AWS Cloudfront HLS CDN
388
204A
202A
Viewer Client Device #1
200A
Client App
5000

FIG. 8

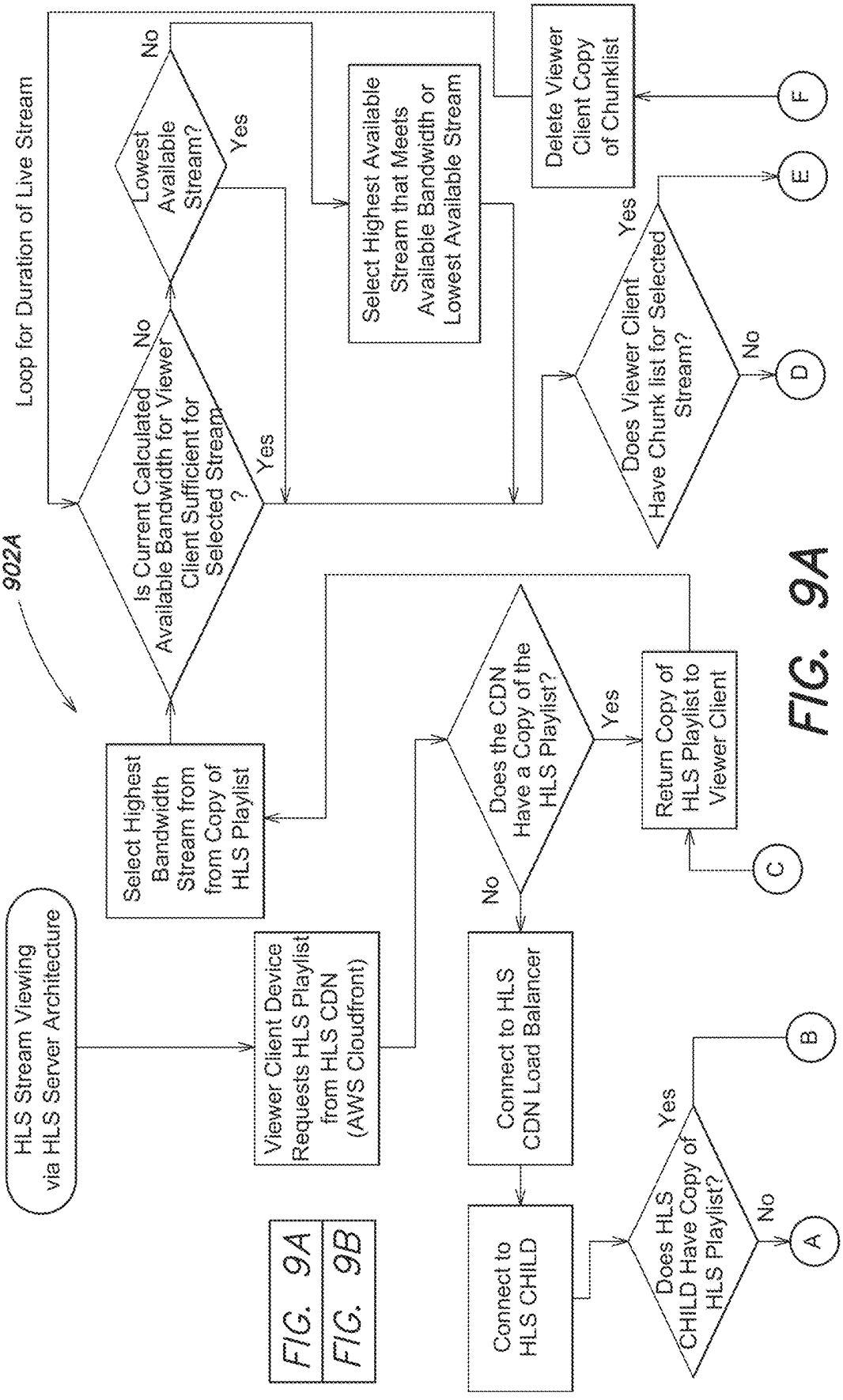


FIG. 9A

FIG. 9A
FIG. 9B

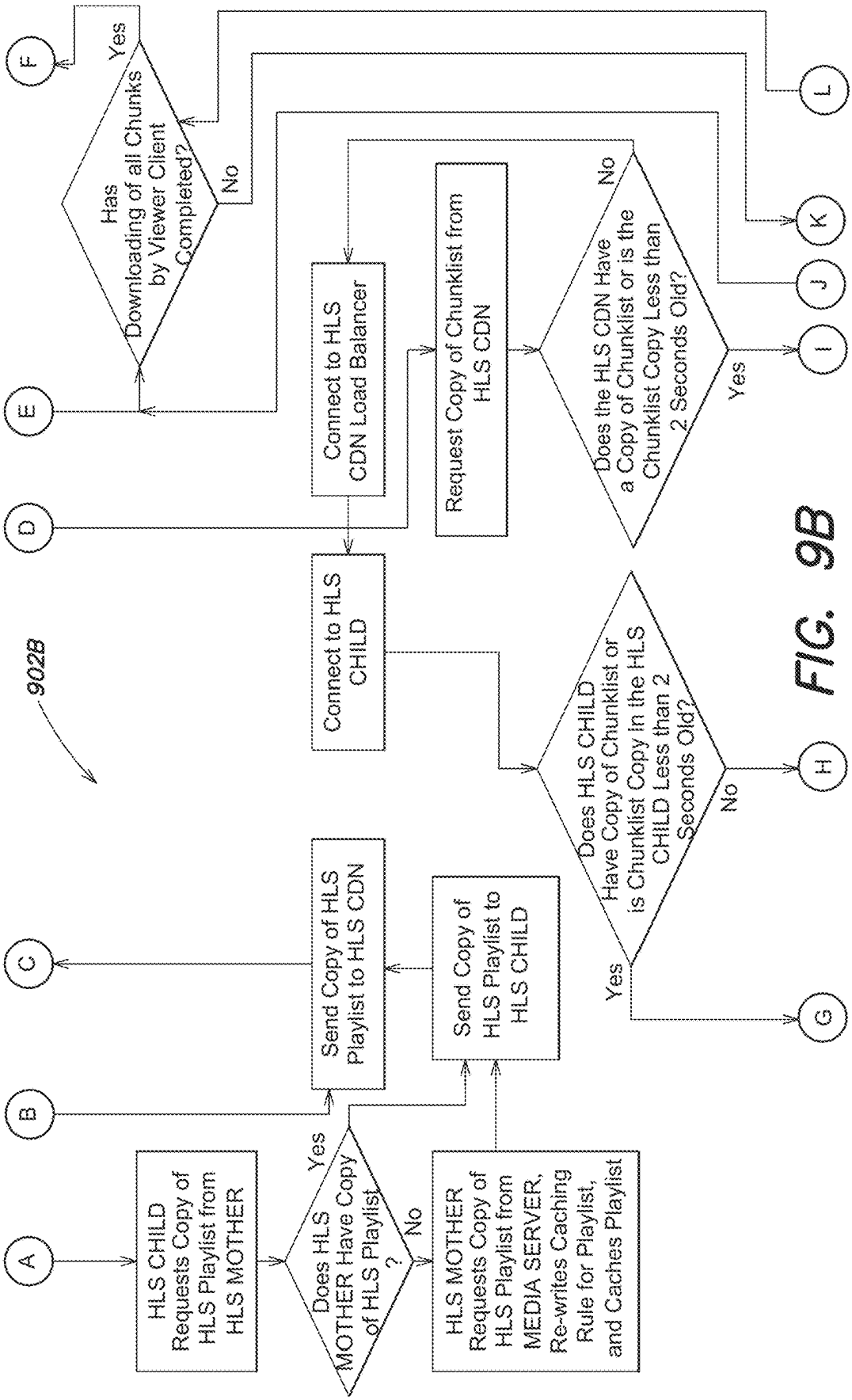

9028
A
HLS CHILD Requests Copy of HLS Playlist from HLS MOTHER
Does HLS MOTHER Have Copy of HLS Playlist ?
Yes
No
HLS MOTHER Requests Copy of HLS Playlist from MEDIA SERVER, Re-writes Caching Rule for Playlist, and Caches Playlist
B
Send Copy of HLS Playlist to HLS CDN
Send Copy of HLS Playlist to HLS CHILD
C
Connect to HLS CHILD
Does HLS CHILD Have Copy of Chunklist or is Chunklist Copy in the HLS CHILD Less than 2 Seconds Old?
Yes
No
G
H
D
Connect to HLS CDN Load Balancer
Request Copy of Chunklist from HLS CDN
Does the HLS CDN Have a Copy of Chunklist or is the Chunklist Copy Less than 2 Seconds Old?
No
Yes
I
J
K
L
E
Has Downloading of all Chunks by Viewer Client Completed?
No
Yes
F

FIG. 9B

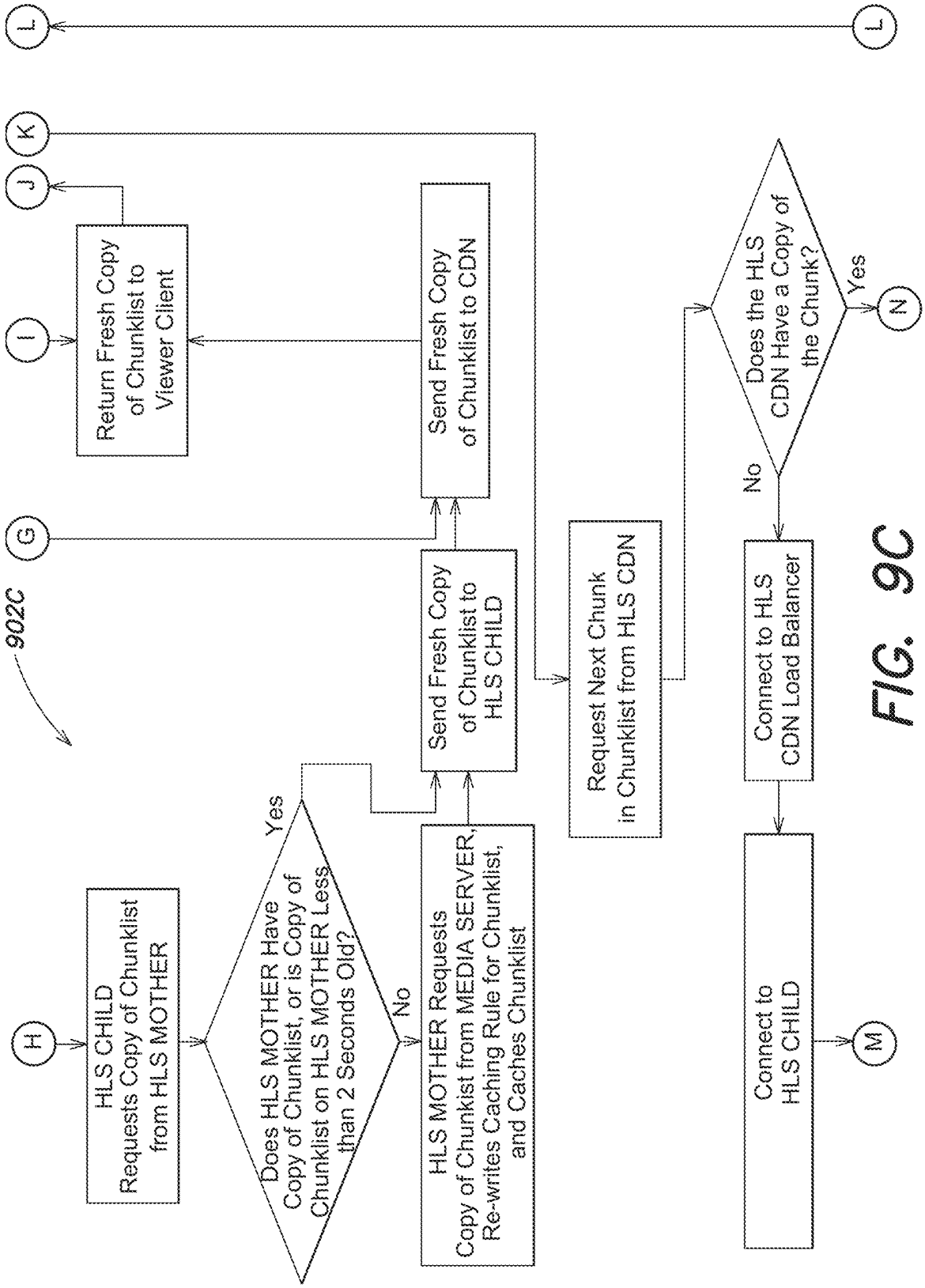

902C
H
HLS CHILD Requests Copy of Chunklist from HLS MOTHER
Does HLS MOTHER Have Copy of Chunklist, or is Copy of Chunklist on HLS MOTHER Less than 2 Seconds Old?
Yes
No
HLS MOTHER Requests Copy of Chunkist from MEDIA SERVER, Re-writes Caching Rule for Chunklist, and Caches Chunklist
Send Fresh Copy of Chunklist to HLS CHILD
G
Send Fresh Copy of Chunklist to CDN
I
Return Fresh Copy of Chunklist to Viewer Client
J
K
L
L
Request Next Chunk in Chunklist from HLS CDN
Does the HLS CDN Have a Copy of the Chunk?
No
Yes
N
Connect to HLS CDN Load Balancer
Connect to HLS CHILD
M

FIG. 9C

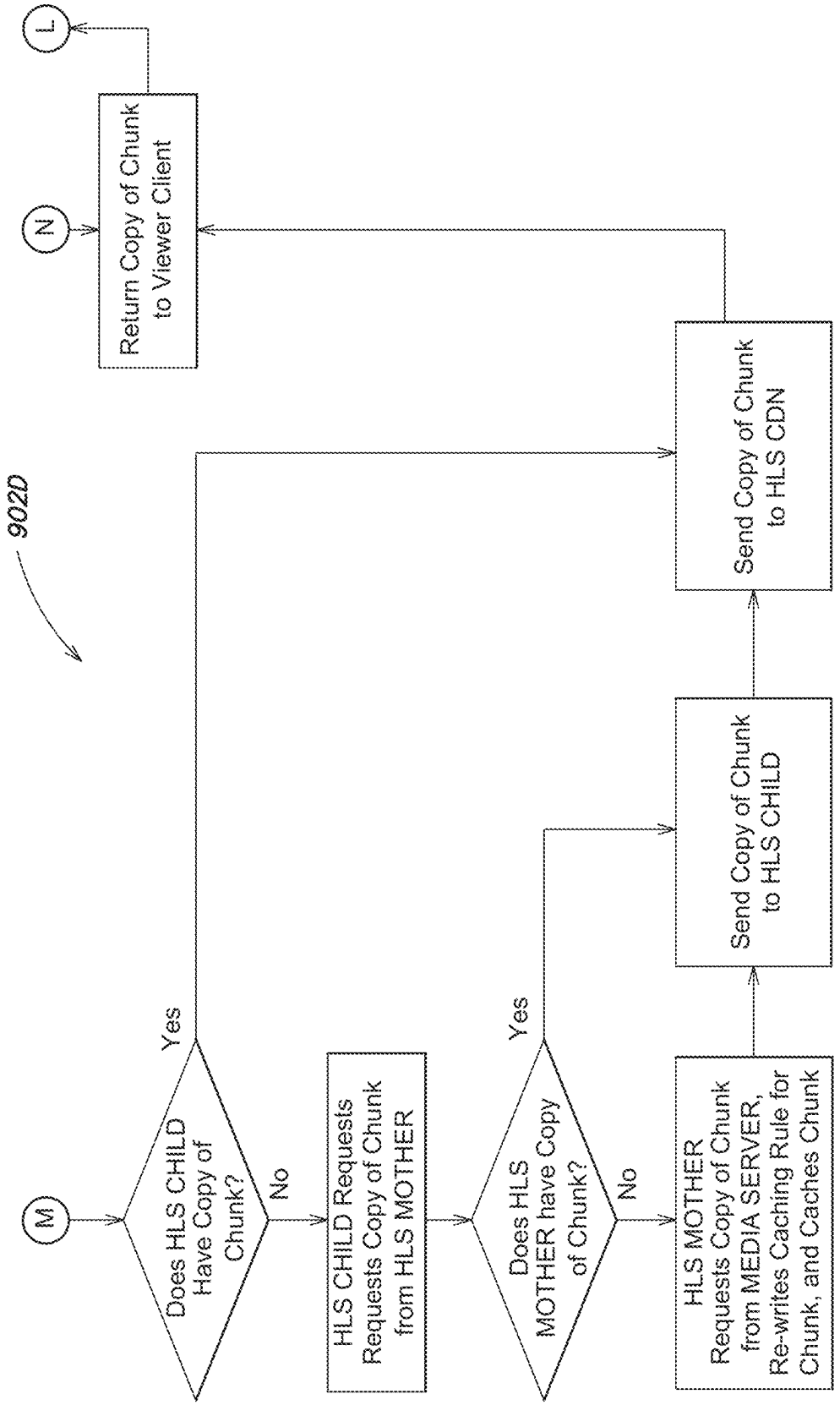

902D
M
Does HLS CHILD Have Copy of Chunk?
Yes
No
HLS CHILD Requests Copy of Chunk from HLS MOTHER
Does HLS MOTHER have Copy of Chunk?
Yes
No
HLS MOTHER Requests Copy of Chunk from MEDIA SERVER, Re-writes Caching Rule for Chunk, and Caches Chunk
Send Copy of Chunk to HLS CHILD
Send Copy of Chunk to HLS CDN
N
Return Copy of Chunk to Viewer Client
L

FIG. 9D

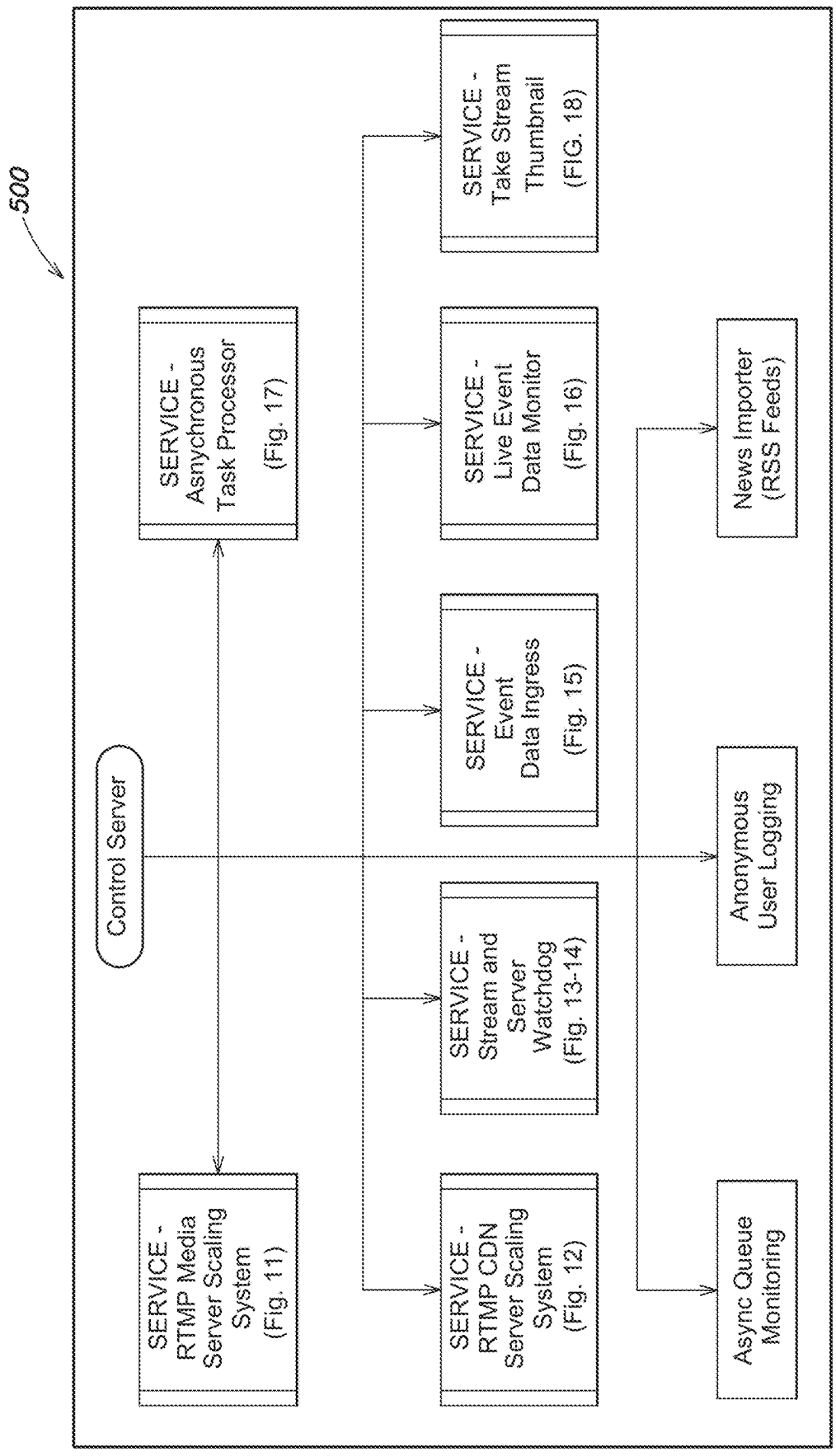

500
Control Server
SERVICE - RTMP Media Server Scaling System (Fig. 11)
SERVICE - Asnychronous Task Processor (Fig. 17)
SERVICE - RTMP CDN Server Scaling System (Fig. 12)
SERVICE - Stream and Server Watchdog (Fig. 13-14)
SERVICE - Event Data Ingress (Fig. 15)
SERVICE - Live Event Data Monitor (Fig. 16)
SERVICE - Take Stream Thumbnail (FIG. 18)
Async Queue Monitoring
Anonymous User Logging
News Importer (RSS Feeds)

FIG. 10

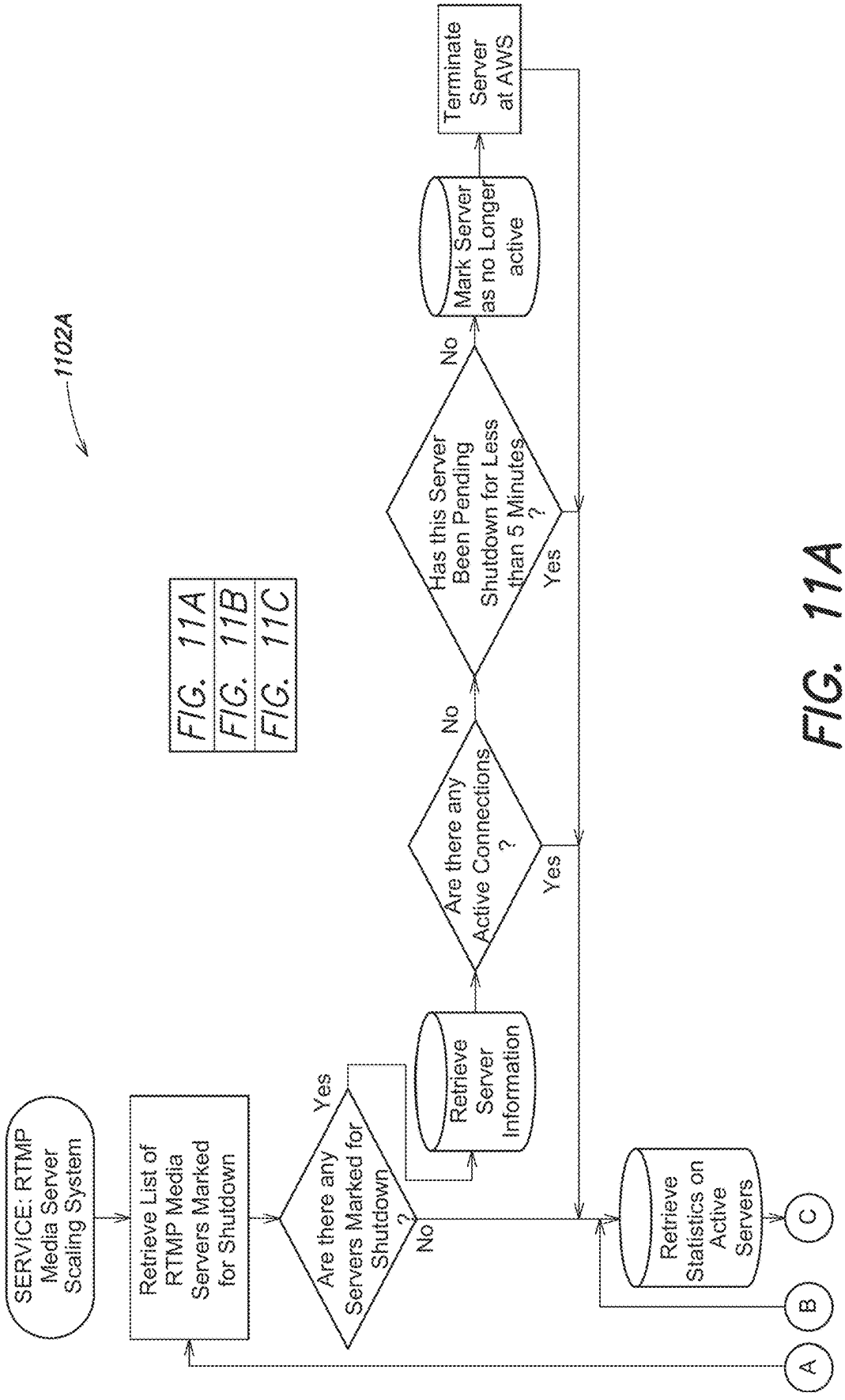


FIG. 11A
FIG. 11B
FIG. 11C

FIG. 11A

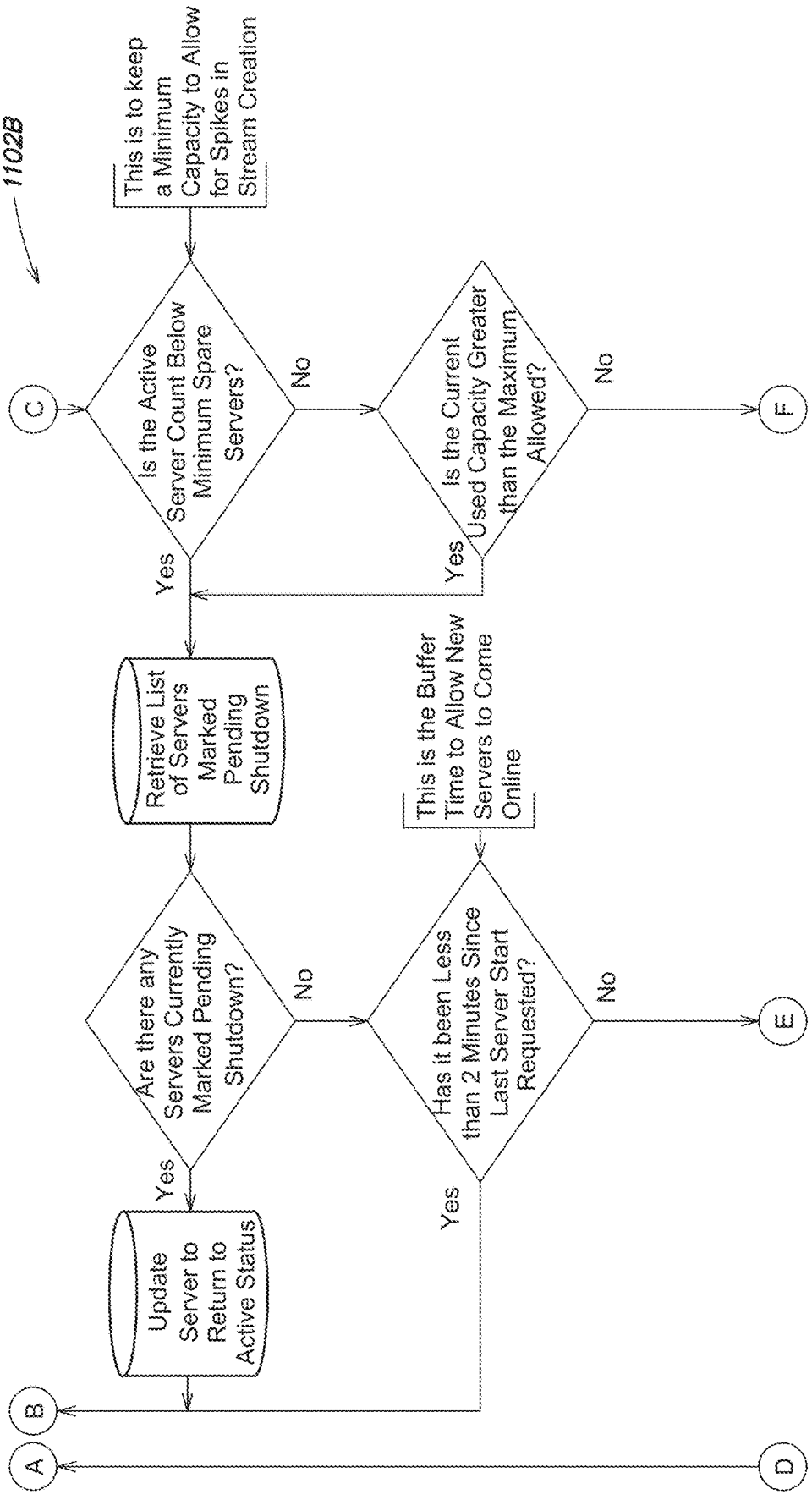

1102B
This is to keep a Minimum Capacity to Allow for Spikes in Stream Creation
C
Is the Active Server Count Below Minimum Spare Servers?
No
Is the Current Used Capacity Greater than the Maximum Allowed?
No
F
Yes
Yes
Retrieve List of Servers Marked Pending Shutdown
This is the Buffer Time to Allow New Servers to Come Online
Are there any Servers Currently Marked Pending Shutdown?
No
Has it been Less than 2 Minutes Since Last Server Start Requested?
No
E
Yes
Yes
Update Server to Return to Active Status
B
A
D

FIG. 11B

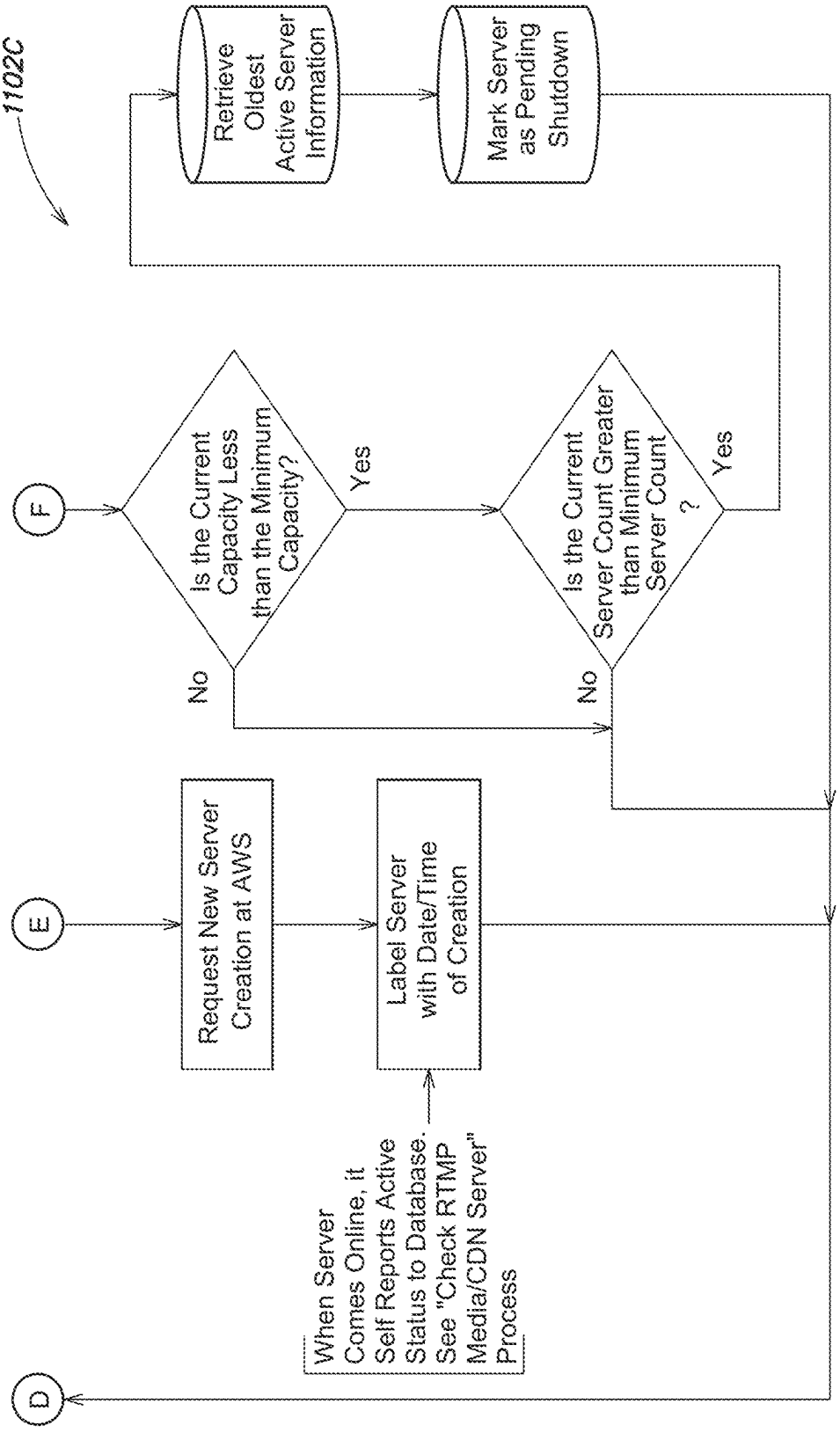

1102C
Retrieve Oldest Active Server Information
Mark Server as Pending Shutdown
F
Is the Current Capacity Less than the Minimum Capacity?
Yes
No
Is the Current Server Count Greater than Minimum Server Count ?
Yes
No
E
Request New Server Creation at AWS
Label Server with Date/Time of Creation
When Server Comes Online, it Self Reports Active Status to Database. See "Check RTMP Media/CDN Server" Process
D

FIG. 11C

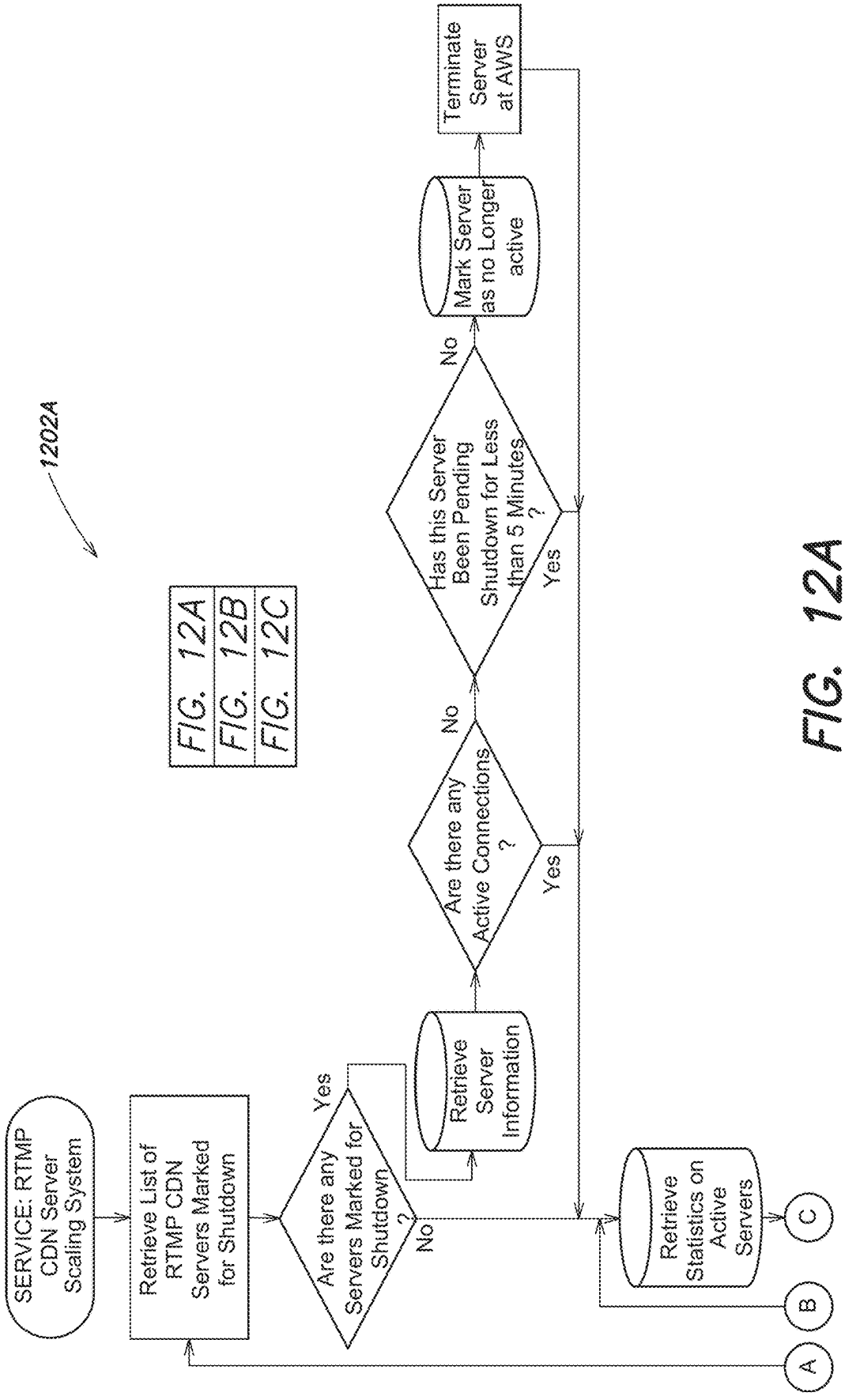


FIG. 12A
FIG. 12B
FIG. 12C

FIG. 12A

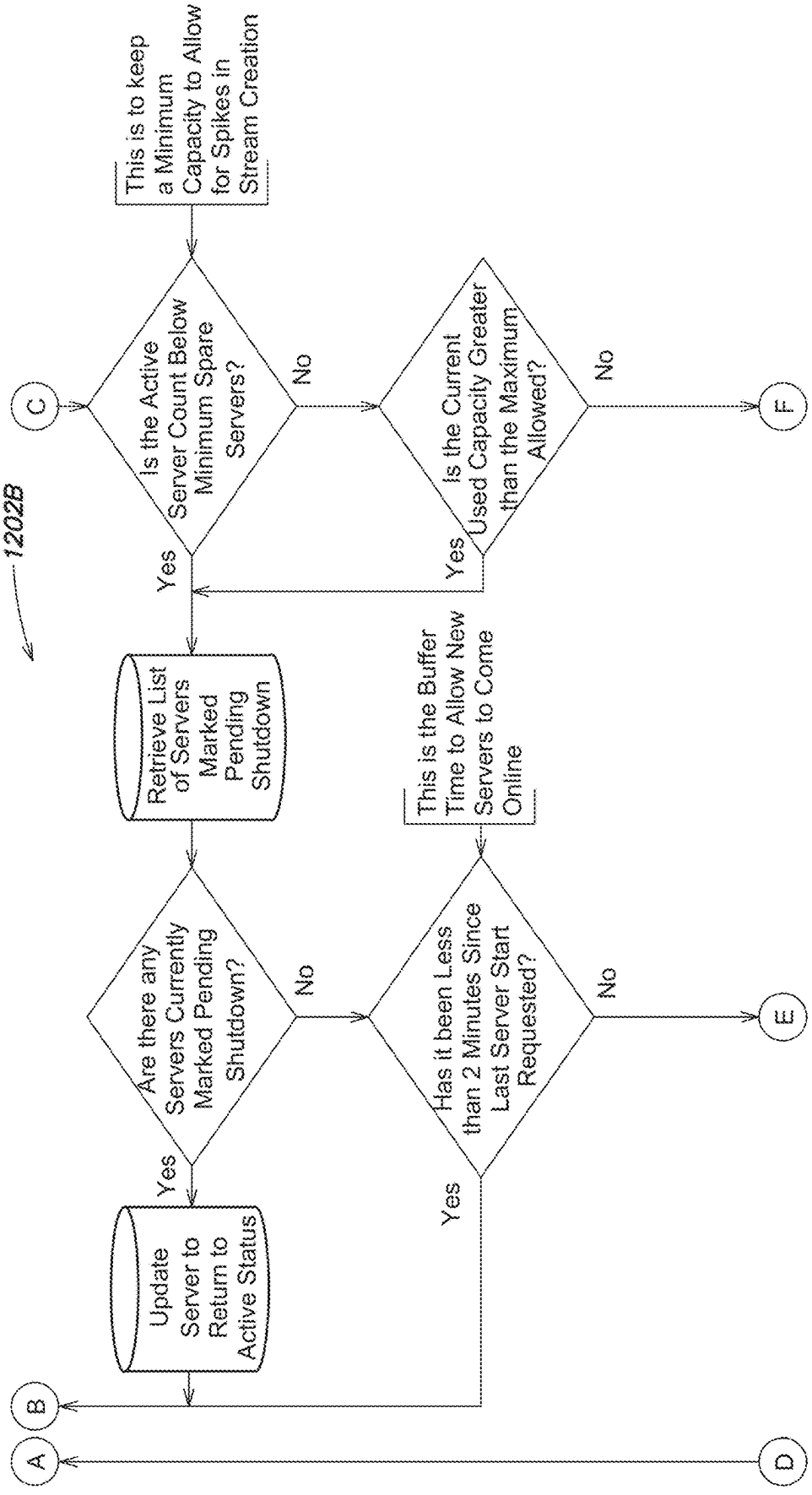

1202B
C
Is the Active Server Count Below Minimum Spare Servers?
No
Is the Current Used Capacity Greater than the Maximum Allowed?
No
F
Yes
Yes
This is to keep a Minimum Capacity to Allow for Spikes in Stream Creation
Retrieve List of Servers Marked Pending Shutdown
Are there any Servers Currently Marked Pending Shutdown?
No
Yes
Update Server to Return to Active Status
This is the Buffer Time to Allow New Servers to Come Online
Has it been Less than 2 Minutes Since Last Server Start Requested?
No
E
Yes
B
A
D

FIG. 12B

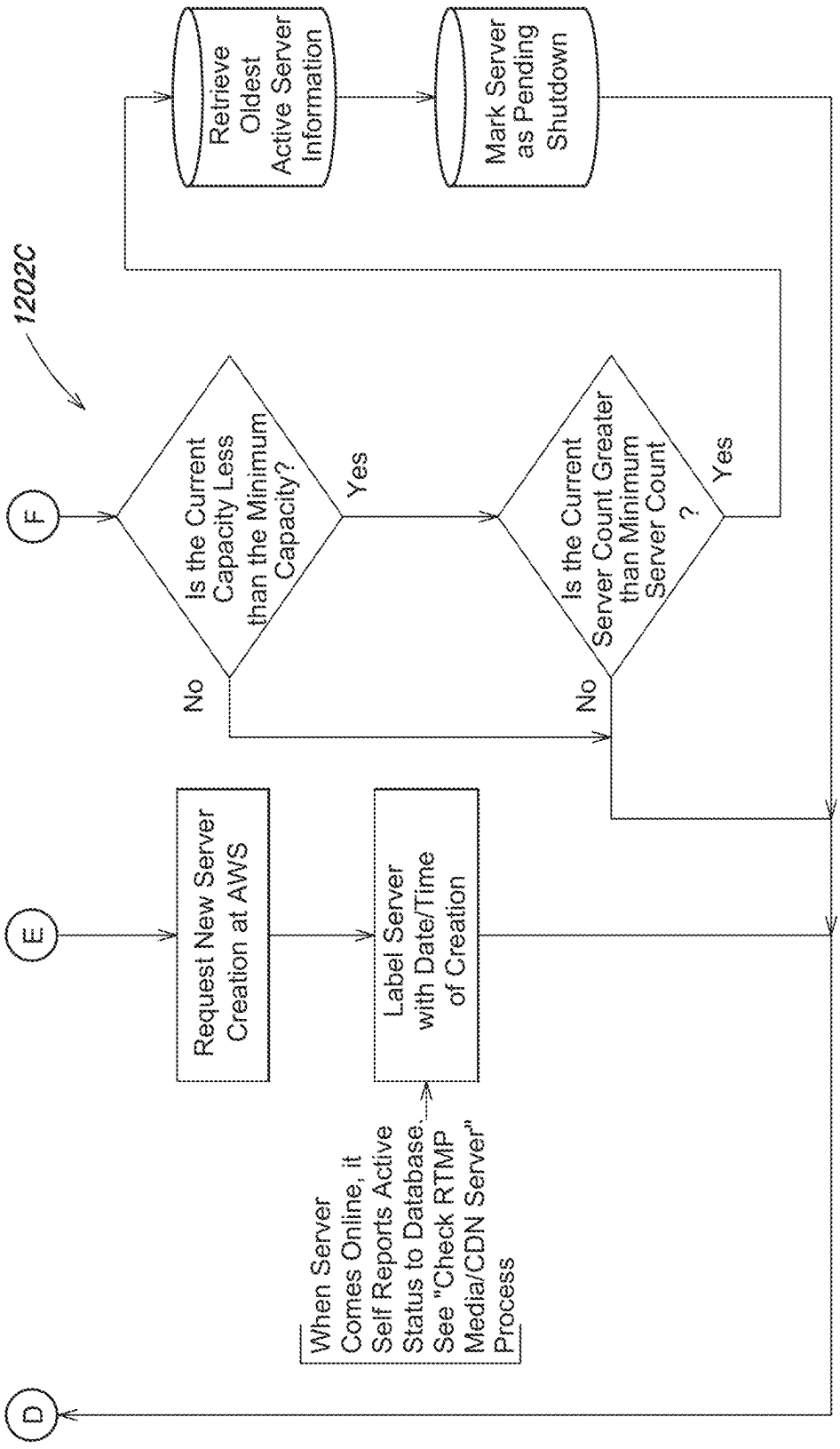

1202C
F
Is the Current Capacity Less than the Minimum Capacity?
Yes
No
Is the Current Server Count Greater than Minimum Server Count ?
Yes
No
Retrieve Oldest Active Server Information
Mark Server as Pending Shutdown
E
Request New Server Creation at AWS
Label Server with Date/Time of Creation
When Server Comes Online, it Self Reports Active Status to Database See "Check RTMP Media/CDN Server" Process
D

FIG. 12C

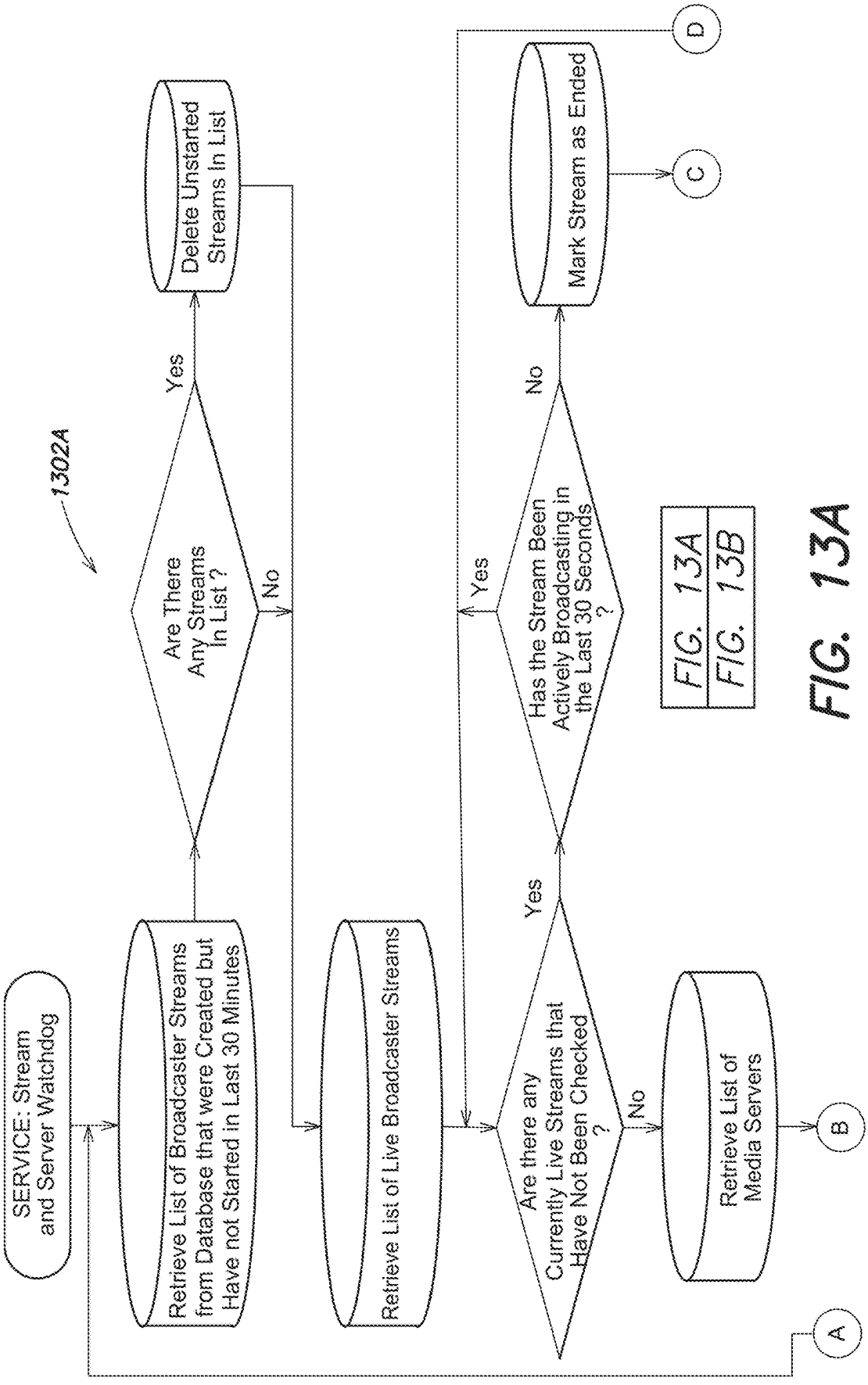


FIG. 13A
FIG. 13B

FIG. 13A

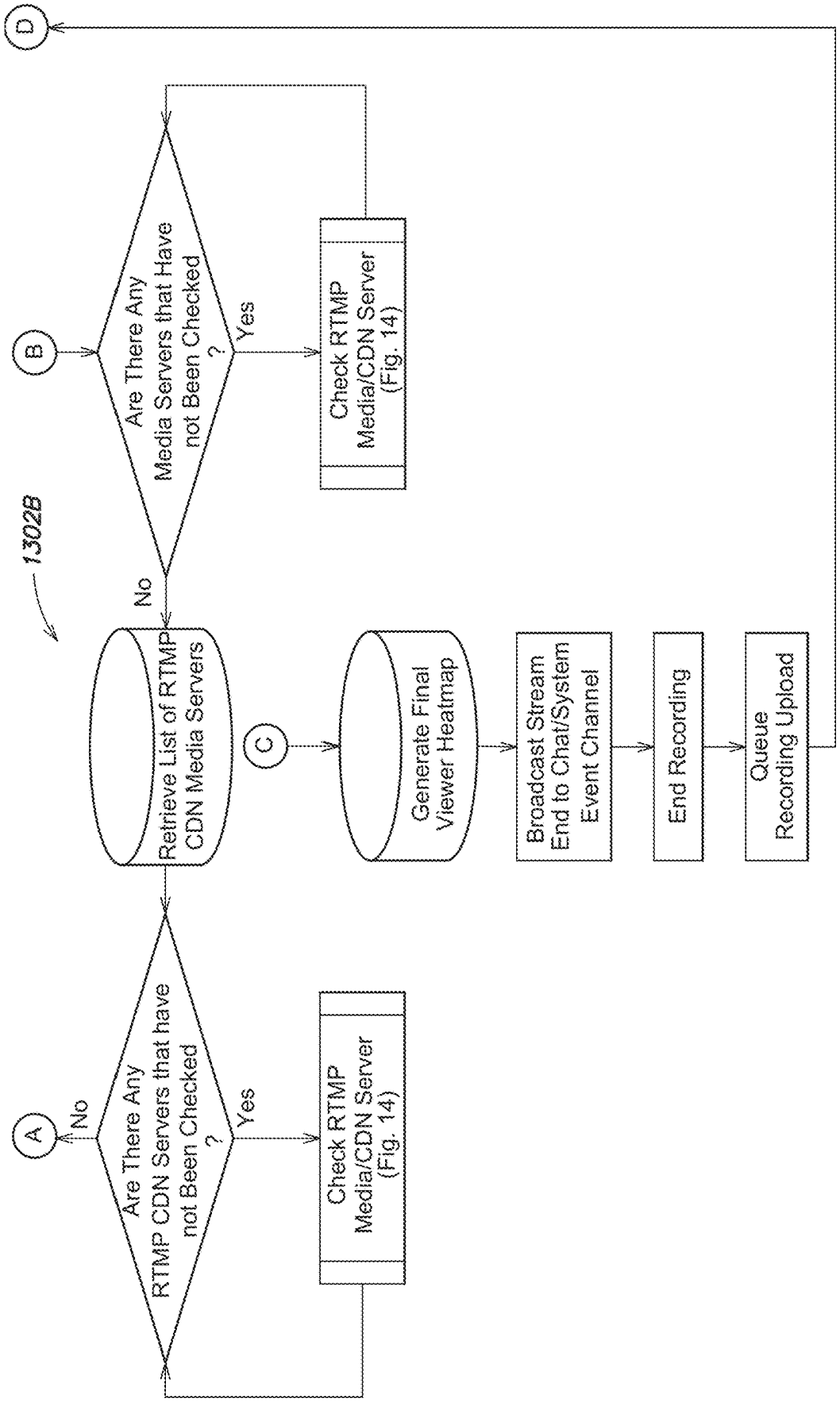

A
No
Are There Any RTMP CDN Servers that have not Been Checked ?
Yes
Check RTMP Media/CDN Server (Fig. 14)
Retrieve List of RTMP CDN Media Servers
B
Are There Any Media Servers that Have not Been Checked ?
Yes
Check RTMP Media/CDN Server (Fig. 14)
No
1302B
C
Generate Final Viewer Heatmap
Broadcast Stream End to Chat/System Event Channel
End Recording
Queue Recording Upload
D

FIG. 13B

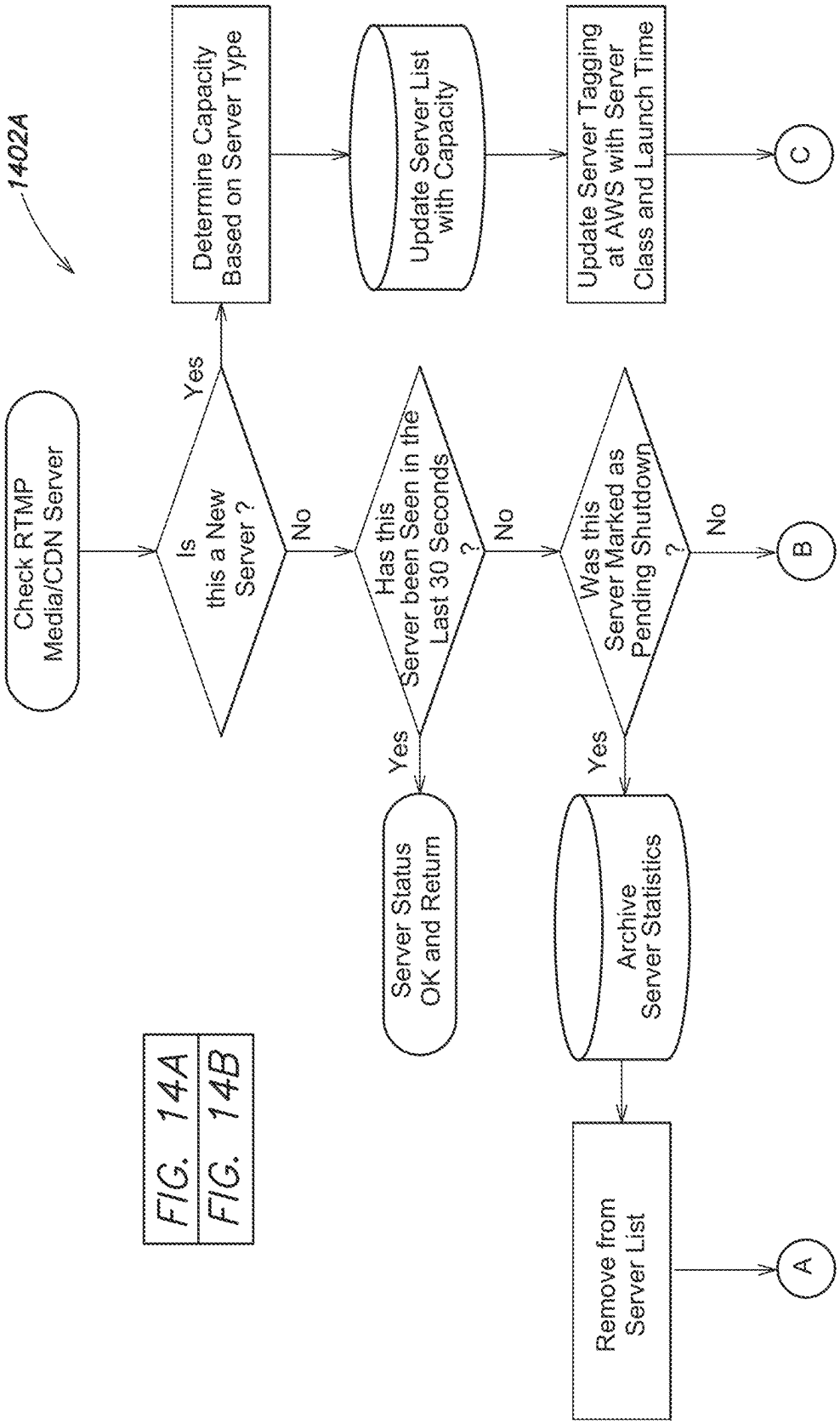


FIG. 14A

FIG. 14A
FIG. 14B

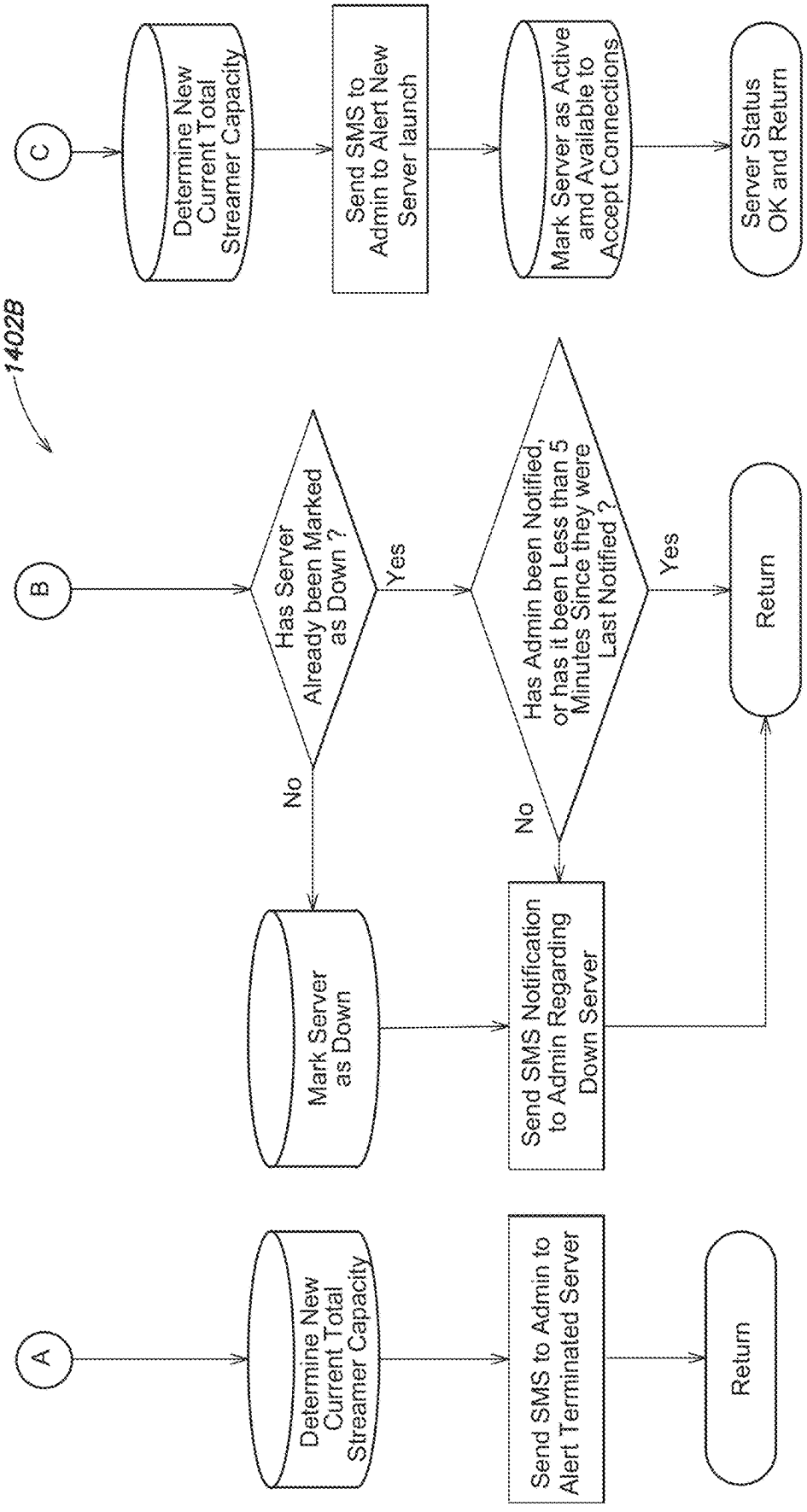

14028
A
B
C
Determine New Current Total Streamer Capacity
Send SMS to Admin to Alert Terminated Server
Return
Has Server Already been Marked as Down ?
Yes
No
Mark Server as Down
Has Admin been Notified, or has it been Less than 5 Minutes Since they were Last Notified ?
Yes
No
Send SMS Notification to Admin Regarding Down Server
Return
Determine New Current Total Streamer Capacity
Send SMS to Admin to Alert New Server launch
Mark Server as Active amd Available to Accept Connections
Server Status OK and Return

FIG. 14B

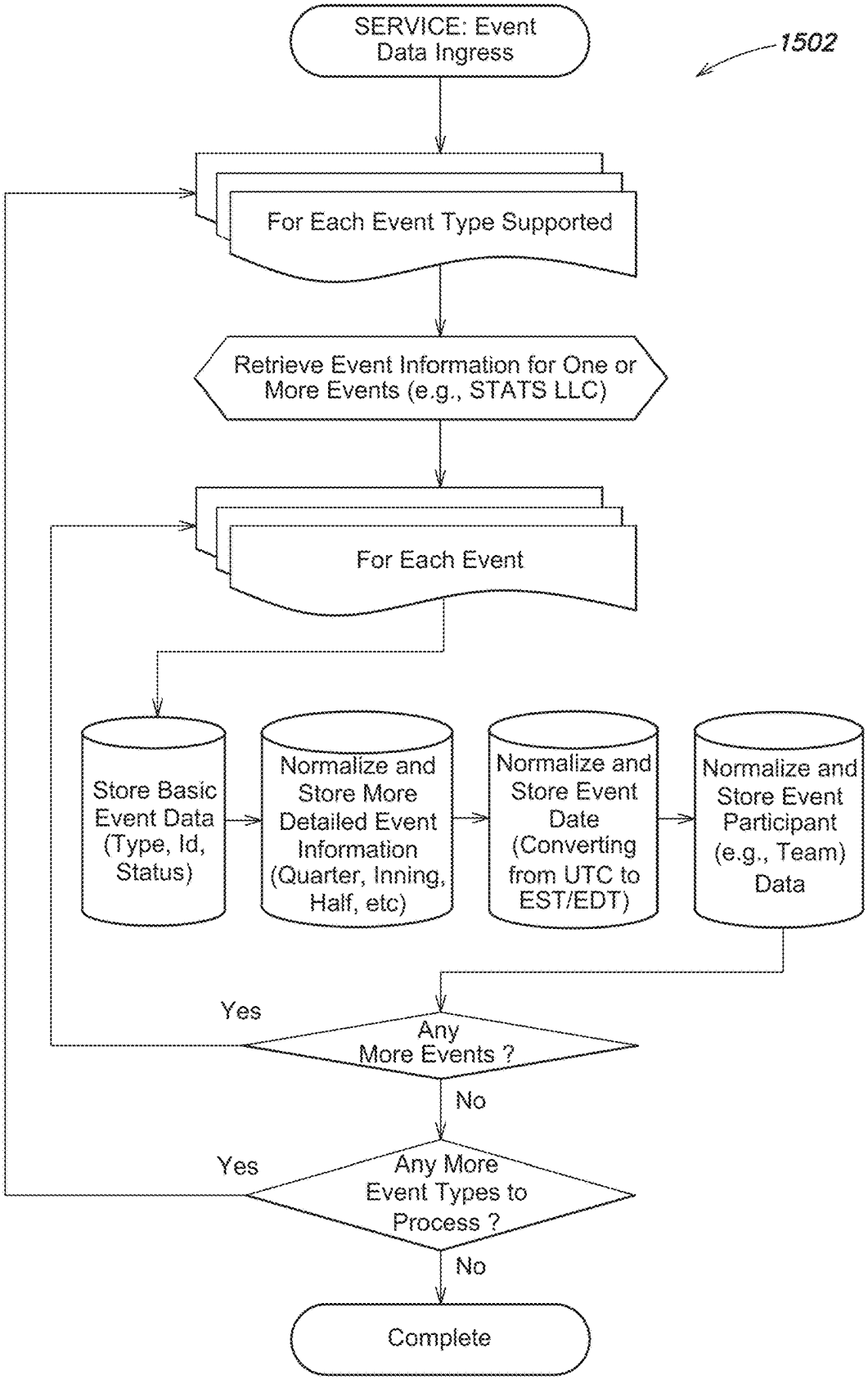

SERVICE: Event Data Ingress
1502
For Each Event Type Supported
Retrieve Event Information for One or More Events (e.g., STATS LLC)
For Each Event
Store Basic Event Data (Type, Id, Status)
Normalize and Store More Detailed Event Information (Quarter, Inning, Half, etc)
Normalize and Store Event Date (Converting from UTC to EST/EDT)
Normalize and Store Event Participant (e.g., Team) Data
Yes
Any More Events ?
No
Yes
Any More Event Types to Process ?
No
Complete

FIG. 15

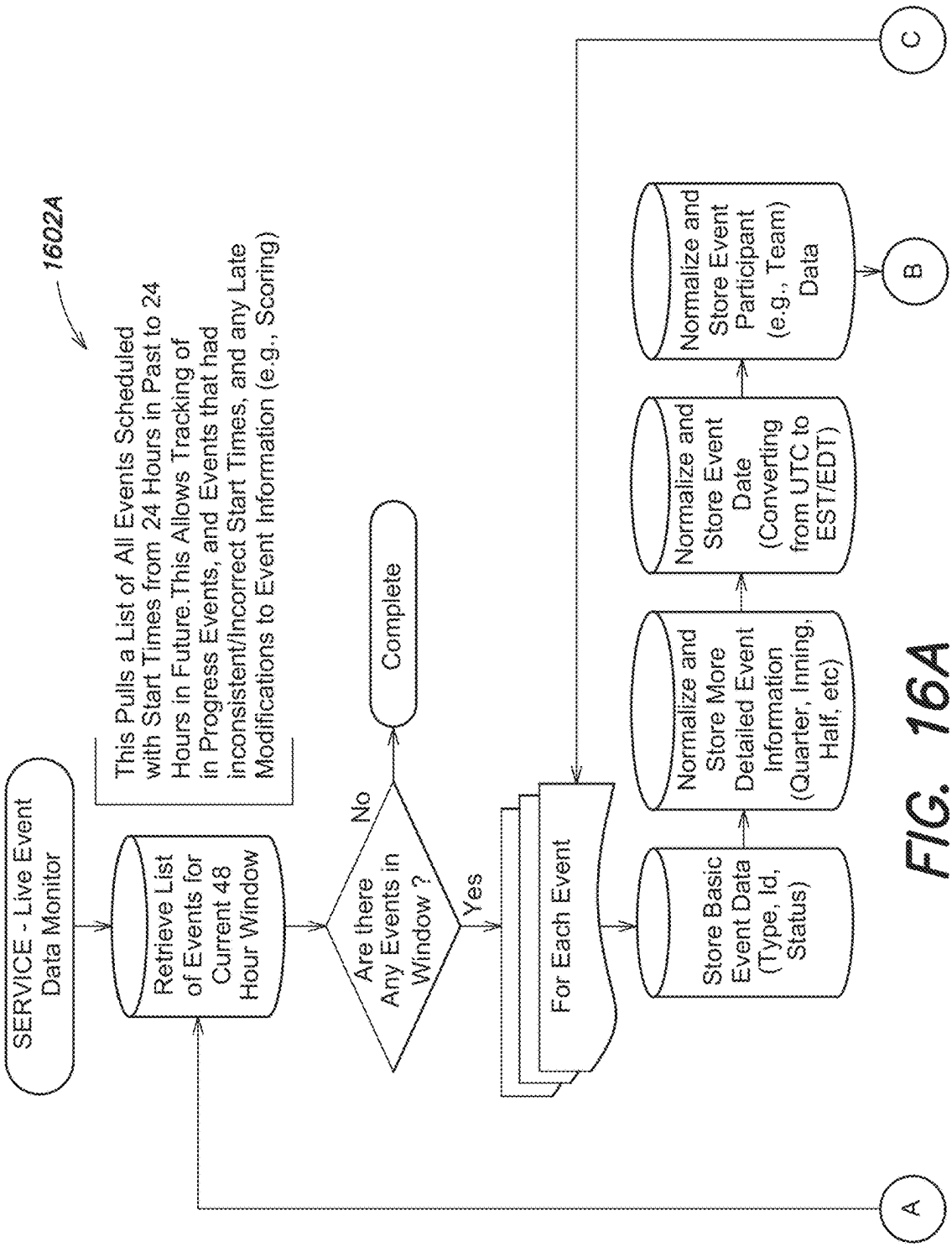


FIG. 16A

FIG. 16A
FIG. 16B

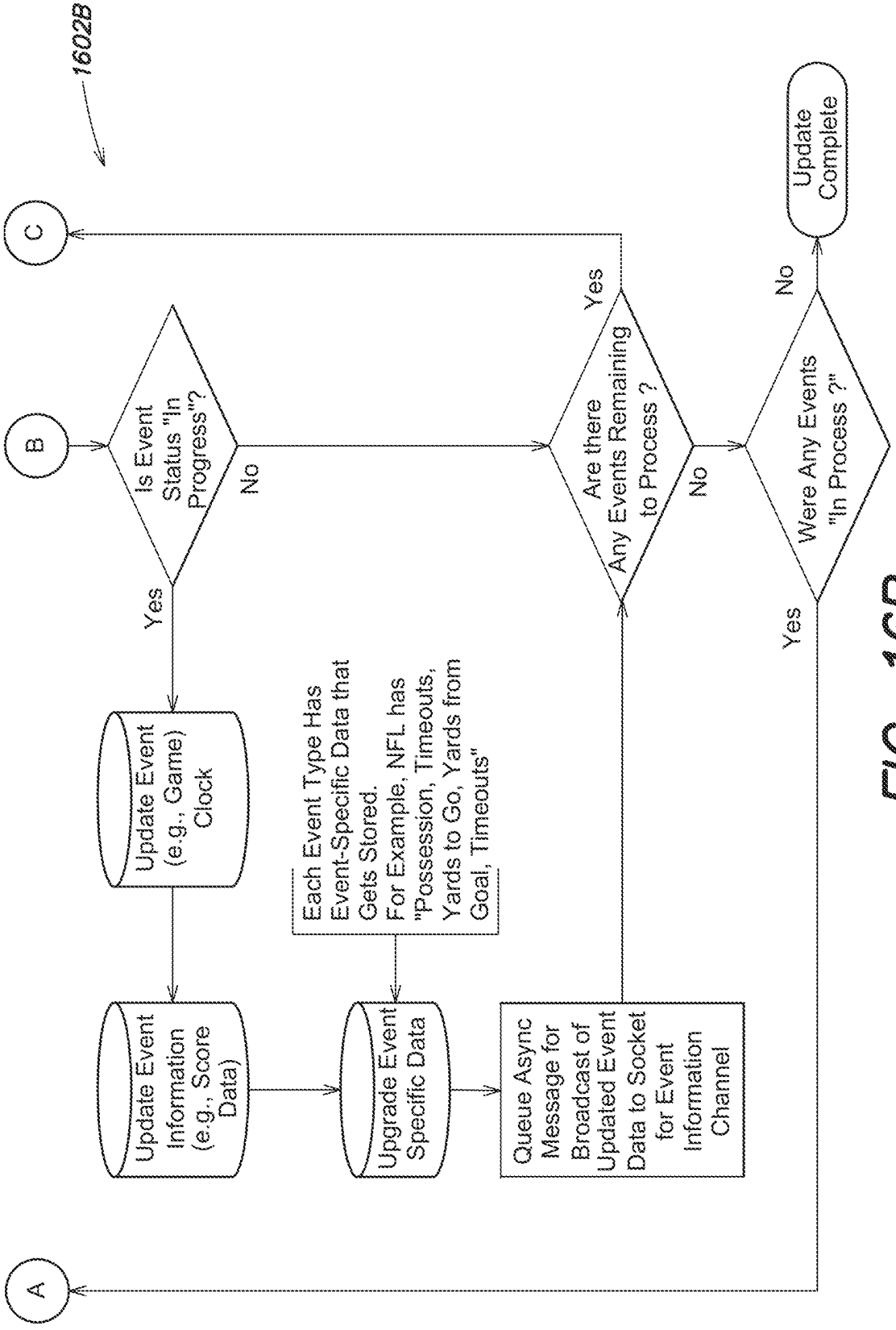

1602B
C
B
A
Is Event Status "In Progress"?
Yes
No
Update Event (e.g., Game) Clock
Update Event Information (e.g., Score Data)
Each Event Type Has Event-Specific Data that Gets Stored.
For Example, NFL has "Possession, Timeouts, Yards to Go, Yards from Goal, Timeouts"
Upgrade Event Specific Data
Queue Async Message for Broadcast of Updated Event Data to Socket for Event Information Channel
Are there Any Events Remaining to Process ?
Yes
No
Were Any Events "In Process ?"
Yes
No
Update Complete

FIG. 16B

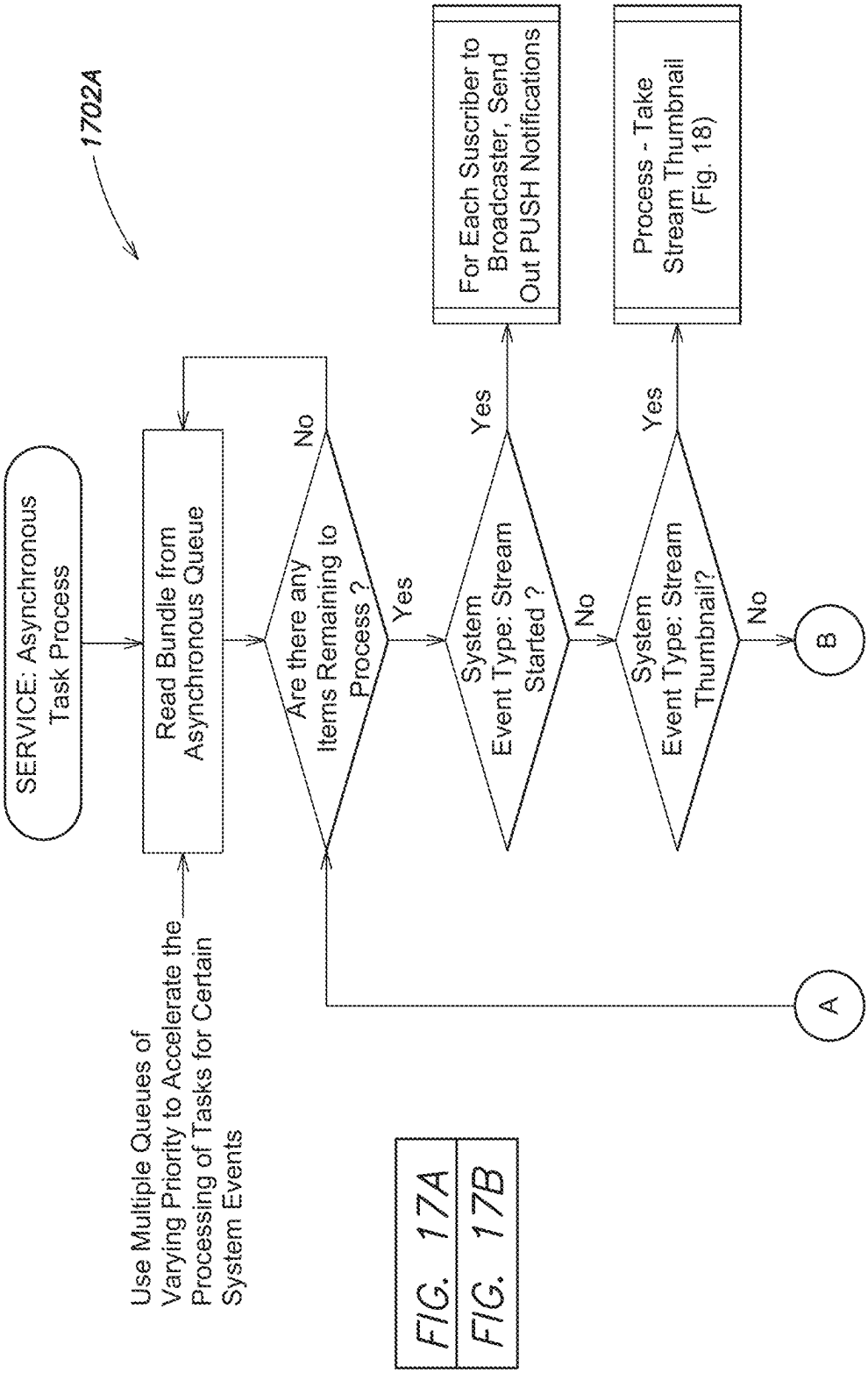


FIG. 17A

FIG. 17A
FIG. 17B

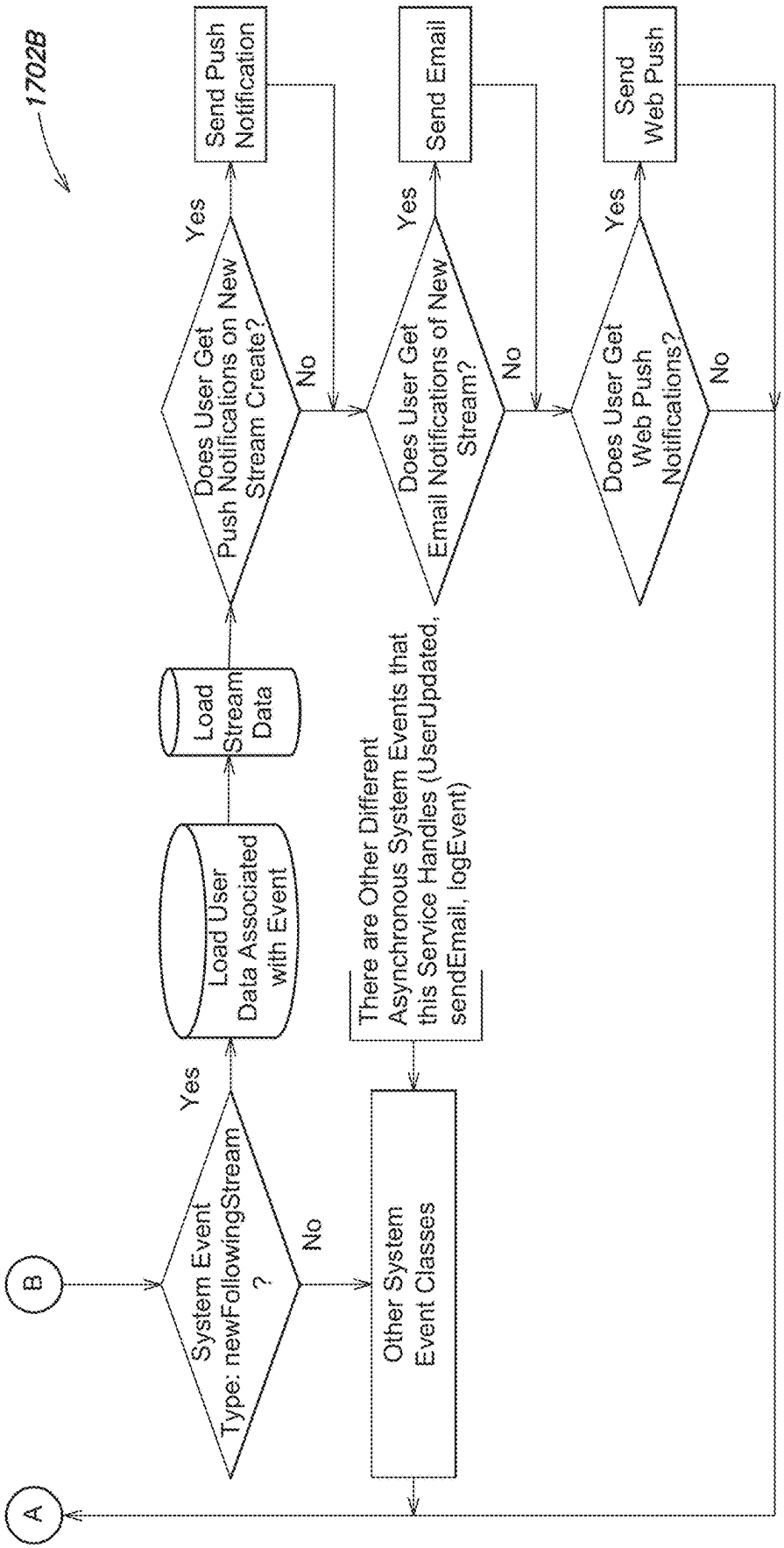

1702B
B
A
System Event Type: newFollowingStream ?
Yes
No
Load User Data Associated with Event
Load Stream Data
There are Other Different Asynchronous System Events that this Service Handles (UserUpdated, sendEmail, logEvent)
Other System Event Classes
Does User Get Push Notifications on New Stream Create?
Yes
No
Send Push Notification
Does User Get Email Notifications of New Stream?
Yes
No
Send Email
Does User Get Web Push Notifications?
Yes
No
Send Web Push

FIG. 17B

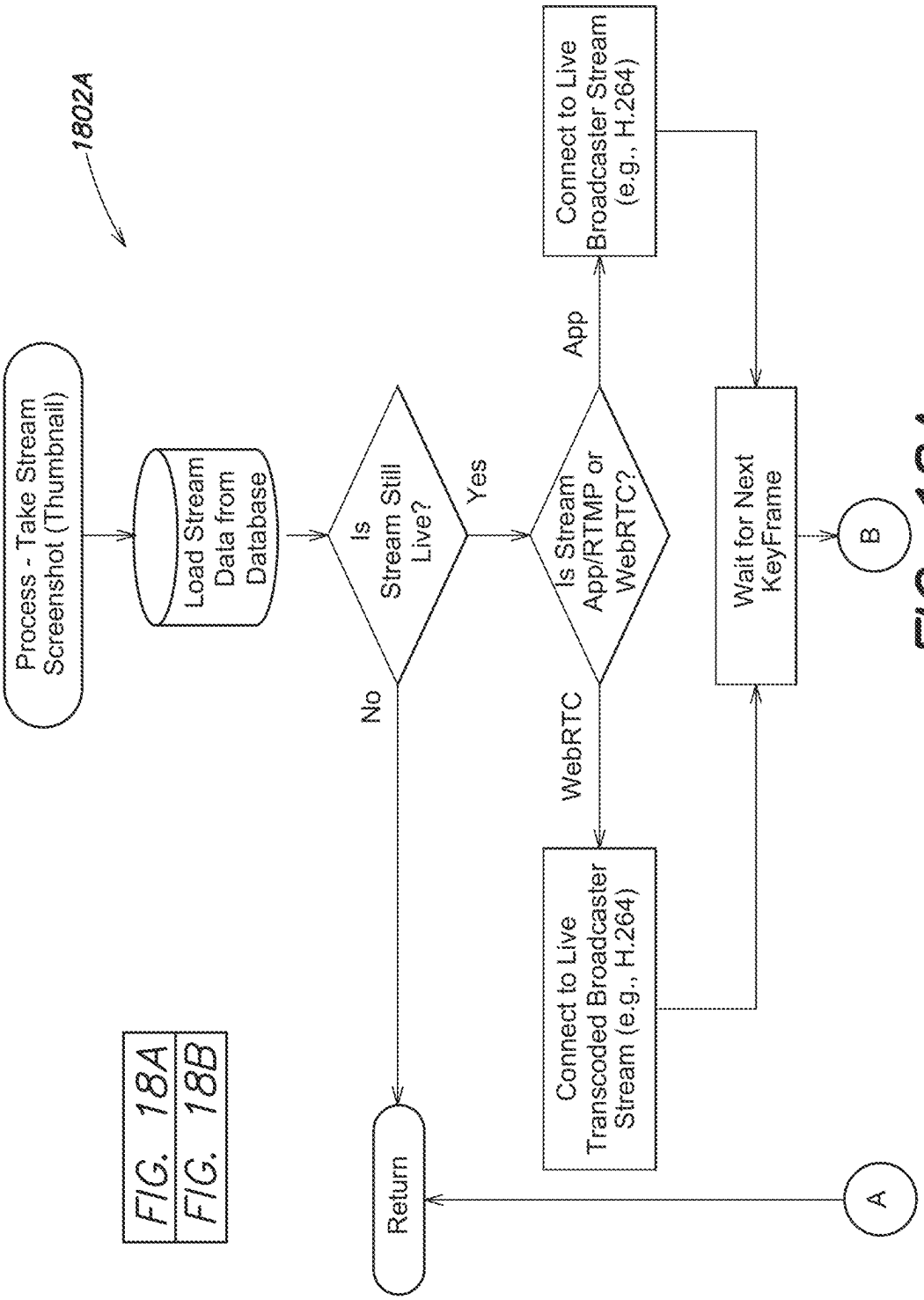


FIG. 18A

FIG. 18A
FIG. 18B

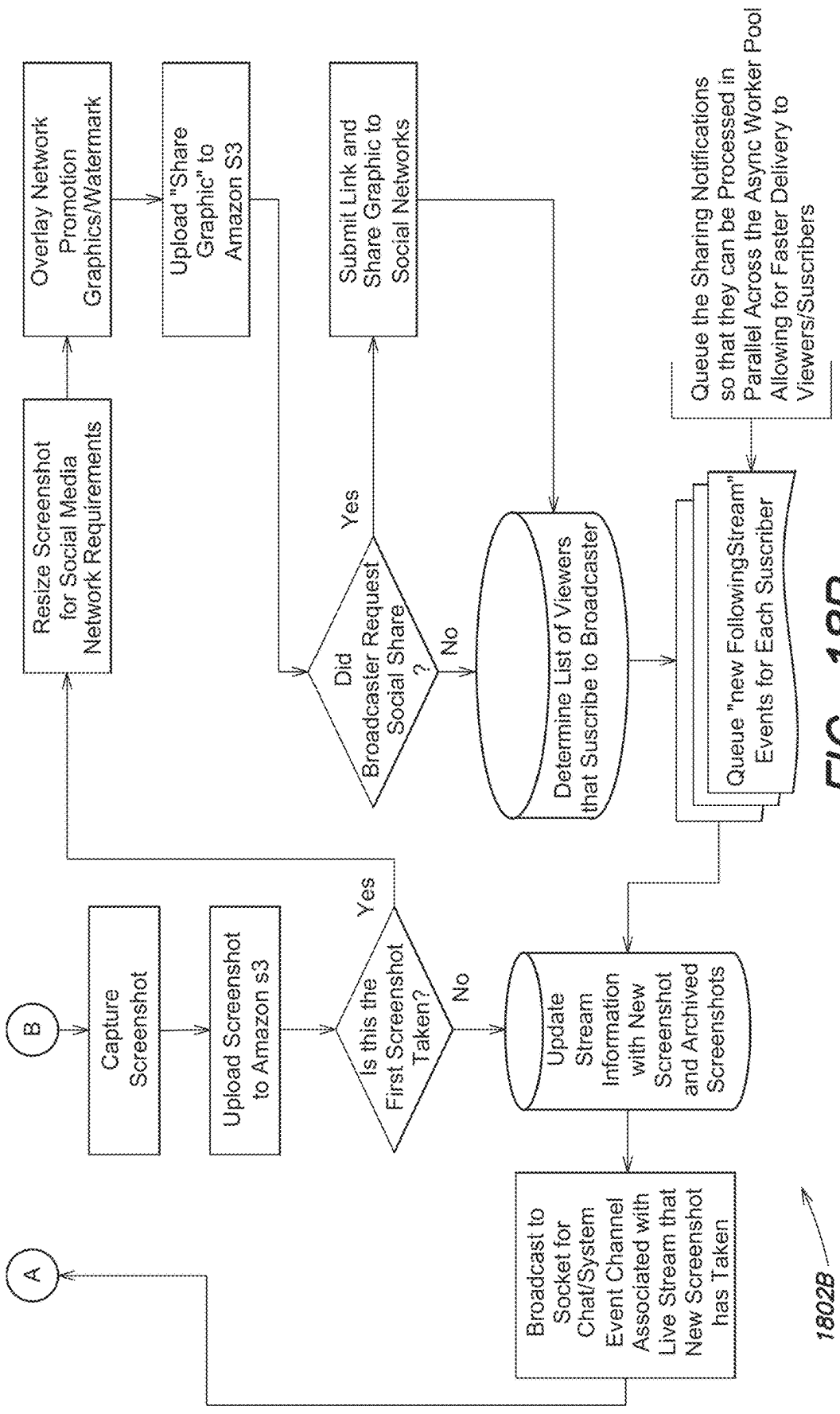

A
B
Capture Screenshot
Upload Screenshot to Amazon s3
Is this the First Screenshot Taken?
Yes
No
Update Stream Information with New Screenshot and Archived Screenshots
Broadcast to Socket for Chat/System Event Channel Associated with Live Stream that New Screenshot has Taken
Resize Screenshot for Social Media Network Requirements
Overlay Network Promotion Graphics/Watermark
Upload "Share Graphic" to Amazon S3
Did Broadcaster Request Social Share ?
Yes
No
Submit Link and Share Graphic to Social Networks
Determine List of Viewers that Suscribe to Broadcaster
Queue "new FollowingStream" Events for Each Suscriber
Queue the Sharing Notifications so that they can be Processed in Parallel Across the Async Worker Pool Allowing for Faster Delivery to Viewers/Suscribers
1802B

FIG. 18B

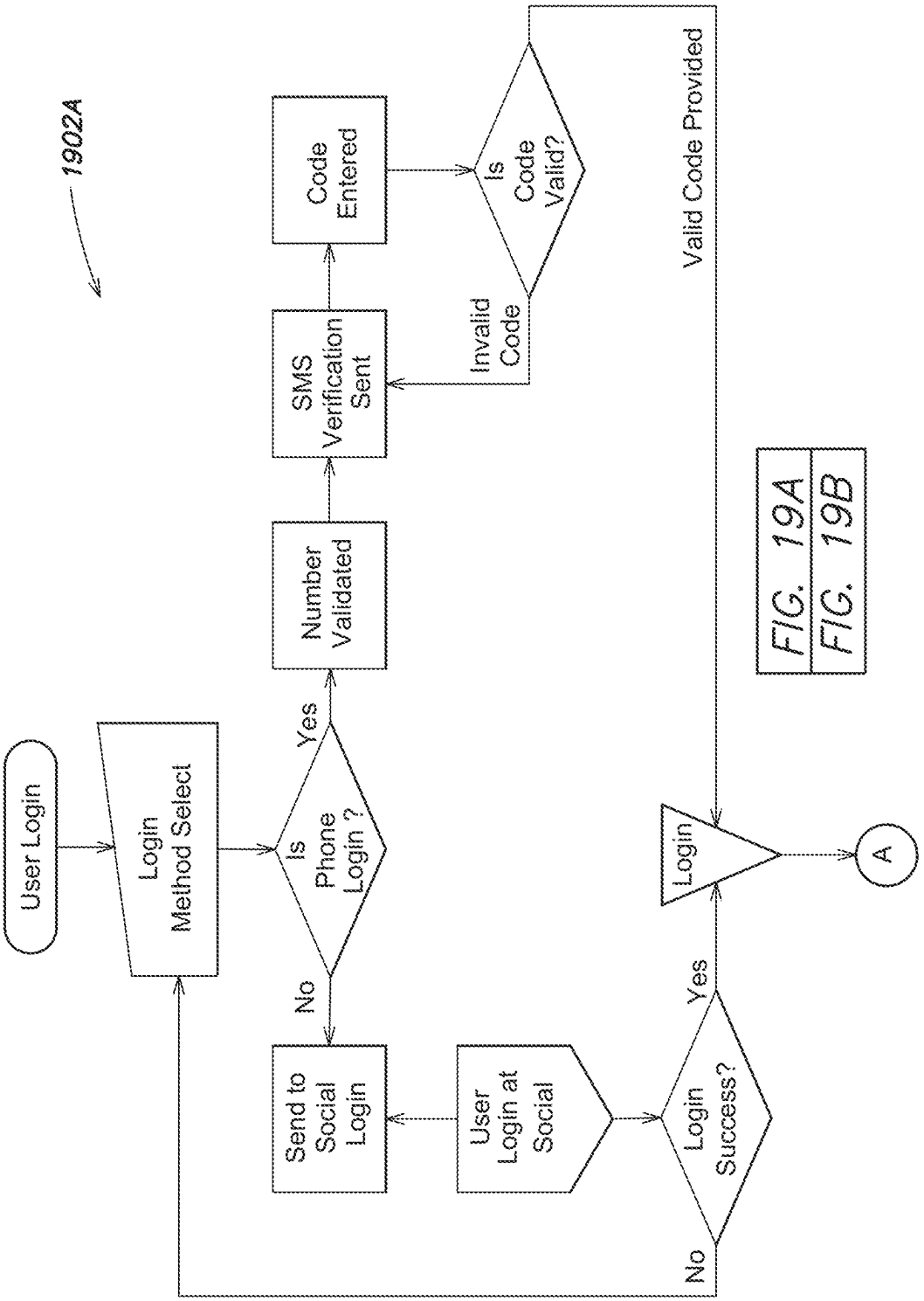


FIG. 19A

FIG. 19A
FIG. 19B

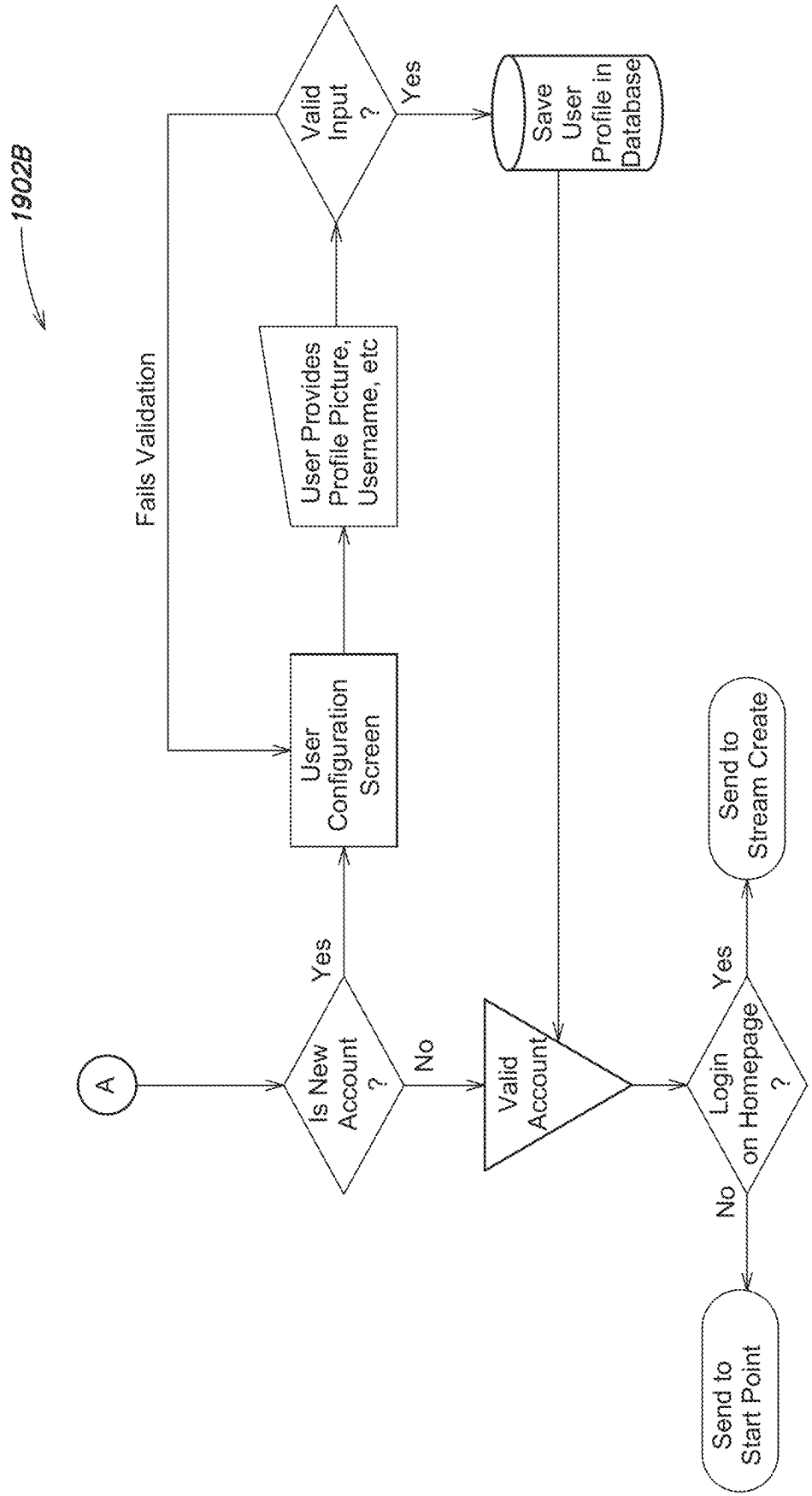

1902B
A
Is New Account ?
Yes
No
User Configuration Screen
User Provides Profile Picture, Username, etc
Valid Input ?
Yes
Fails Validation
Save User Profile in Database
Valid Account
Login on Homepage ?
Yes
No
Send to Stream Create
Send to Start Point

FIG. 19B

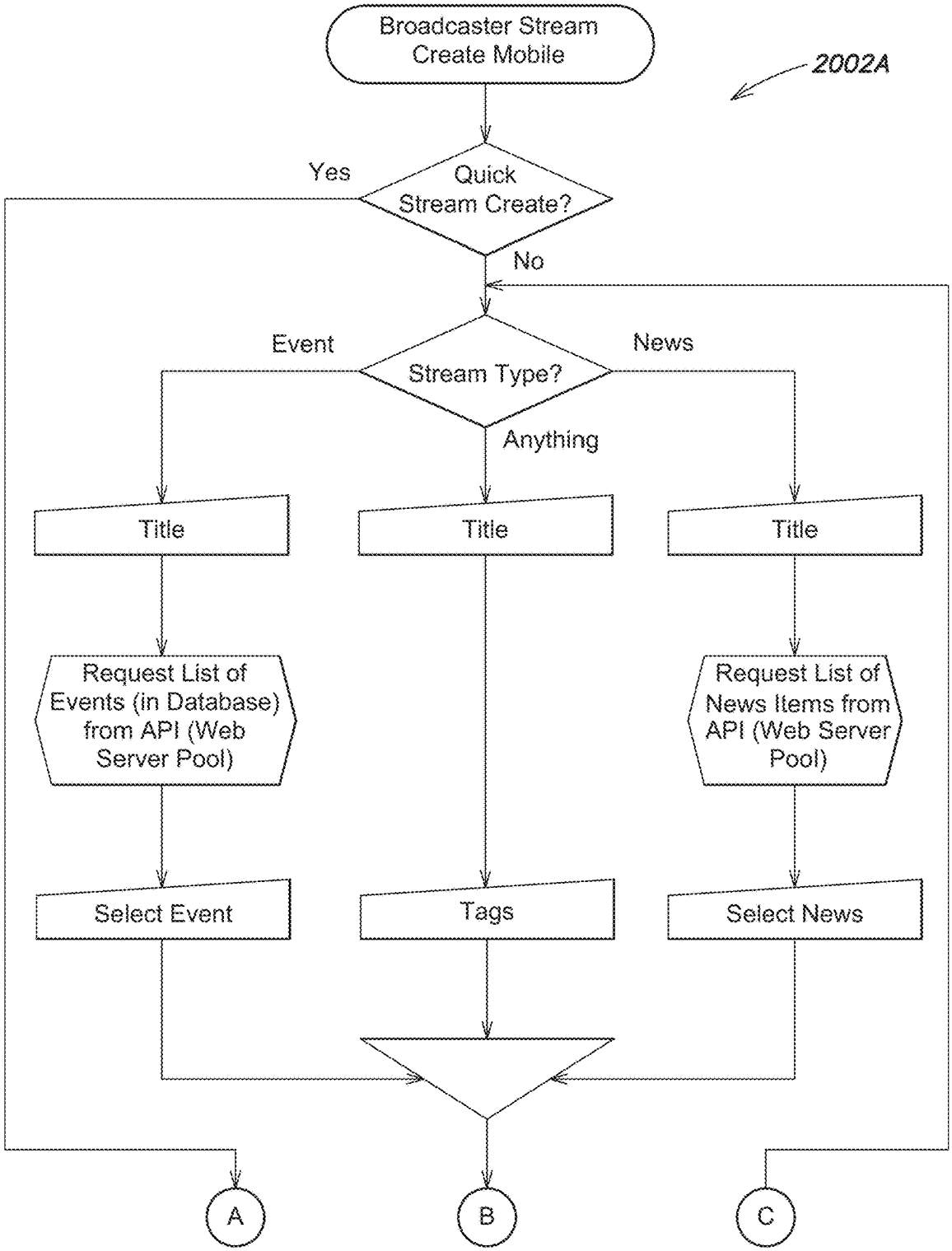


FIG. 20A
FIG. 20B

FIG. 20A

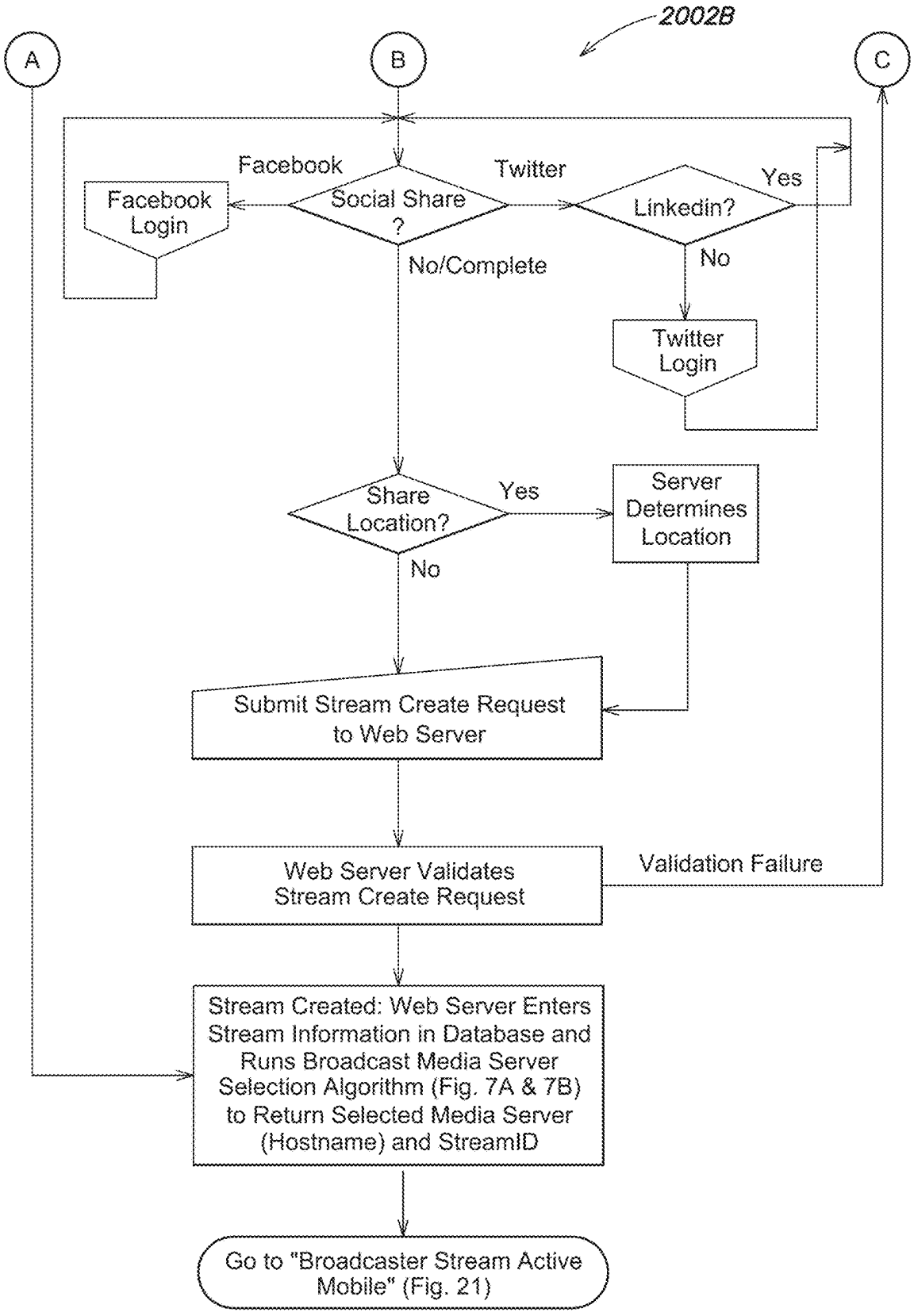

2002B
A
B
C
Facebook
Twitter
Yes
Facebook Login
Social Share ?
Linkedin?
No/Complete
No
Twitter Login
Share Location?
Yes
Server Determines Location
No
Submit Stream Create Request to Web Server
Web Server Validates Stream Create Request
Validation Failure
Stream Created: Web Server Enters Stream Information in Database and Runs Broadcast Media Server Selection Algorithm (Fig. 7A & 7B) to Return Selected Media Server (Hostname) and StreamID
Go to "Broadcaster Stream Active Mobile" (Fig. 21)

FIG. 20B

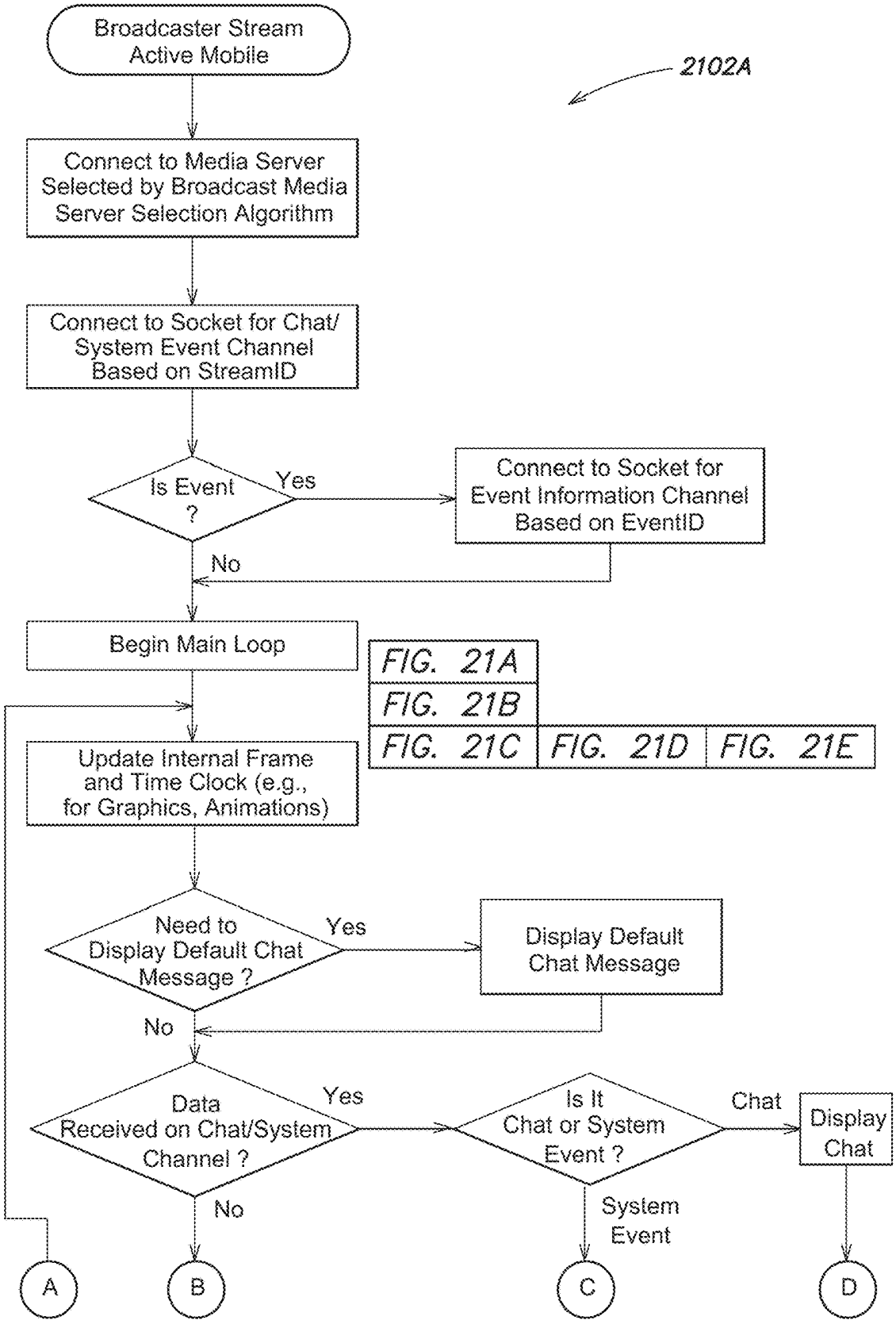

Broadcaster Stream Active Mobile
2102A
Connect to Media Server Selected by Broadcast Media Server Selection Algorithm
Connect to Socket for Chat/ System Event Channel Based on StreamID
Is Event ?
Yes
Connect to Socket for Event Information Channel Based on EventID
No
Begin Main Loop
FIG. 21A
FIG. 21B
FIG. 21C
FIG. 21D
FIG. 21E
Update Internal Frame and Time Clock (e.g., for Graphics, Animations)
Need to Display Default Chat Message ?
Yes
Display Default Chat Message
No
Data Received on Chat/System Channel ?
Yes
Is It Chat or System Event ?
Chat
Display Chat
No
System Event
A
B
C
D

FIG. 21A

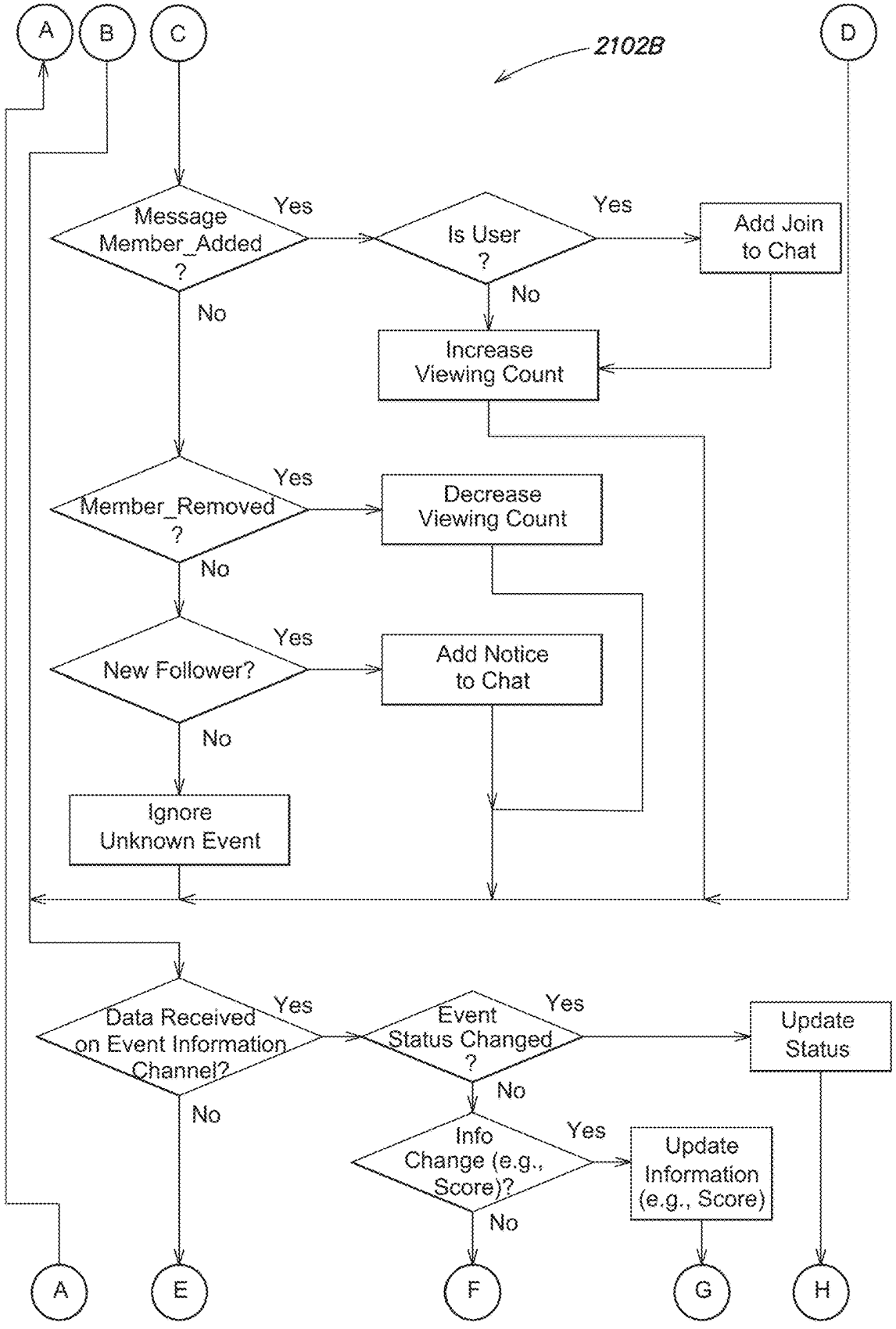

A
B
C
D
2102B
Message Member_Added ?
Yes
No
Is User ?
Yes
No
Add Join to Chat
Increase Viewing Count
Member_Removed ?
Yes
No
Decrease Viewing Count
New Follower?
Yes
No
Add Notice to Chat
Ignore Unknown Event
Data Received on Event Information Channel?
Yes
No
Event Status Changed ?
Yes
No
Update Status
Info Change (e.g., Score)?
Yes
No
Update Information (e.g., Score)
A
E
F
G
H

FIG. 21B

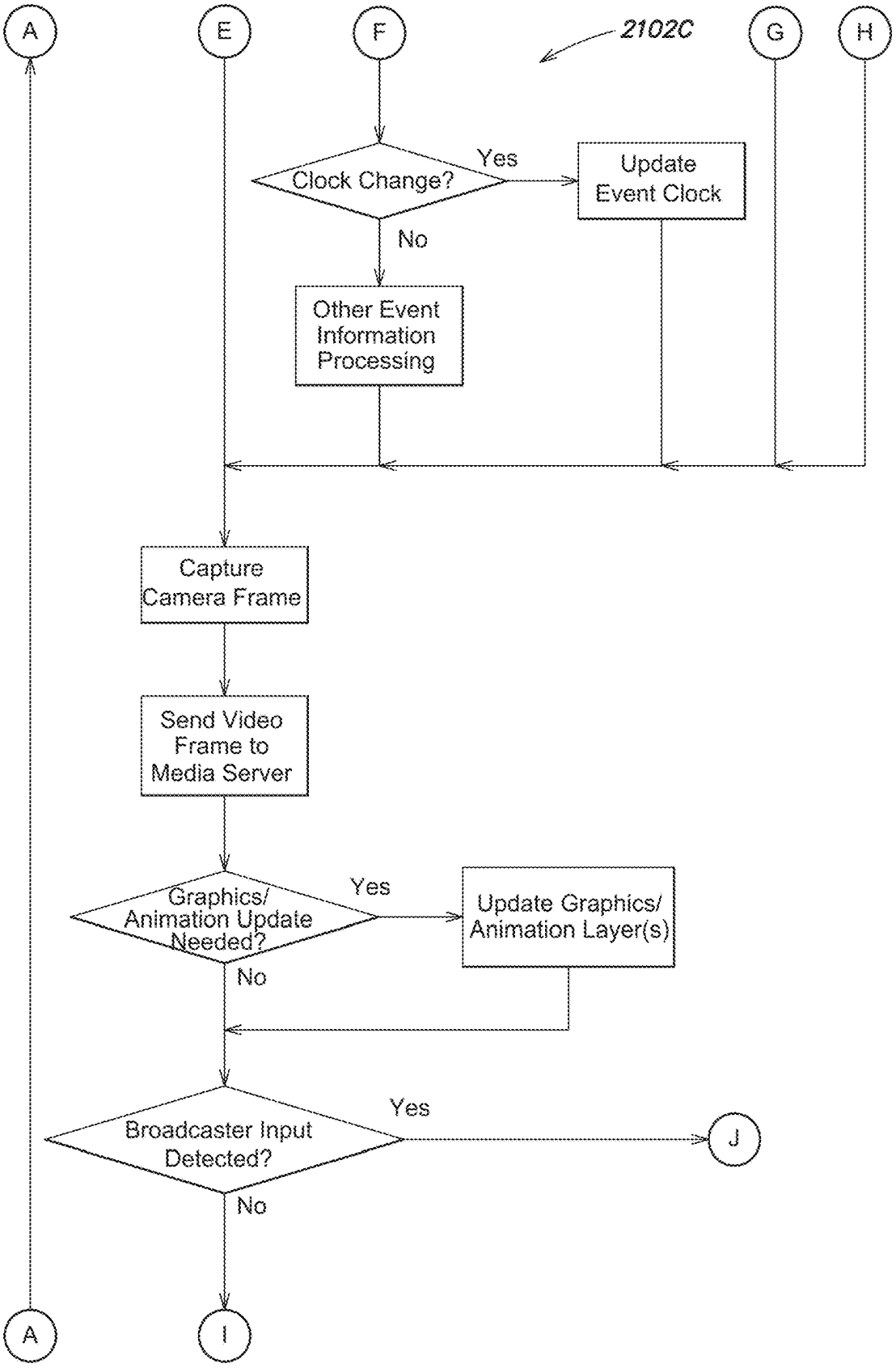

A
E
F
2102C
G
H
Clock Change?
Yes
Update Event Clock
No
Other Event Information Processing
Capture Camera Frame
Send Video Frame to Media Server
Graphics/ Animation Update Needed?
Yes
Update Graphics/ Animation Layer(s)
No
Broadcaster Input Detected?
Yes
J
No
A
I

FIG. 21C

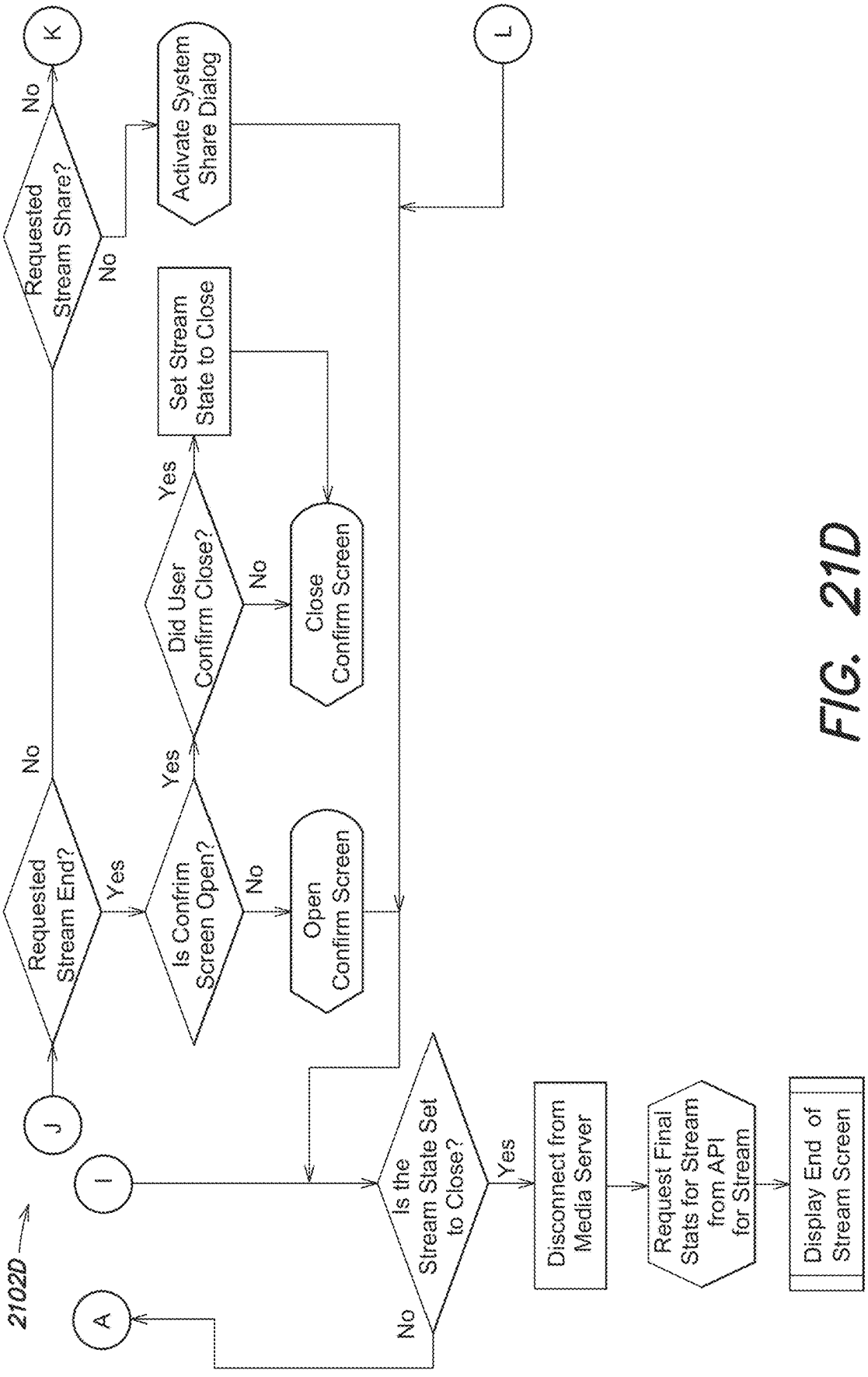

2102D
A
I
J
K
L
Is the Stream State Set to Close?
No
Yes
Disconnect from Media Server
Request Final Stats for Stream from API for Stream
Display End of Stream Screen
Requested Stream End?
No
Yes
Is Confirm Screen Open?
No
Yes
Open Confirm Screen
Did User Confirm Close?
No
Yes
Close Confirm Screen
Set Stream State to Close
Requested Stream Share?
No
Yes
Activate System Share Dialog

FIG. 21D

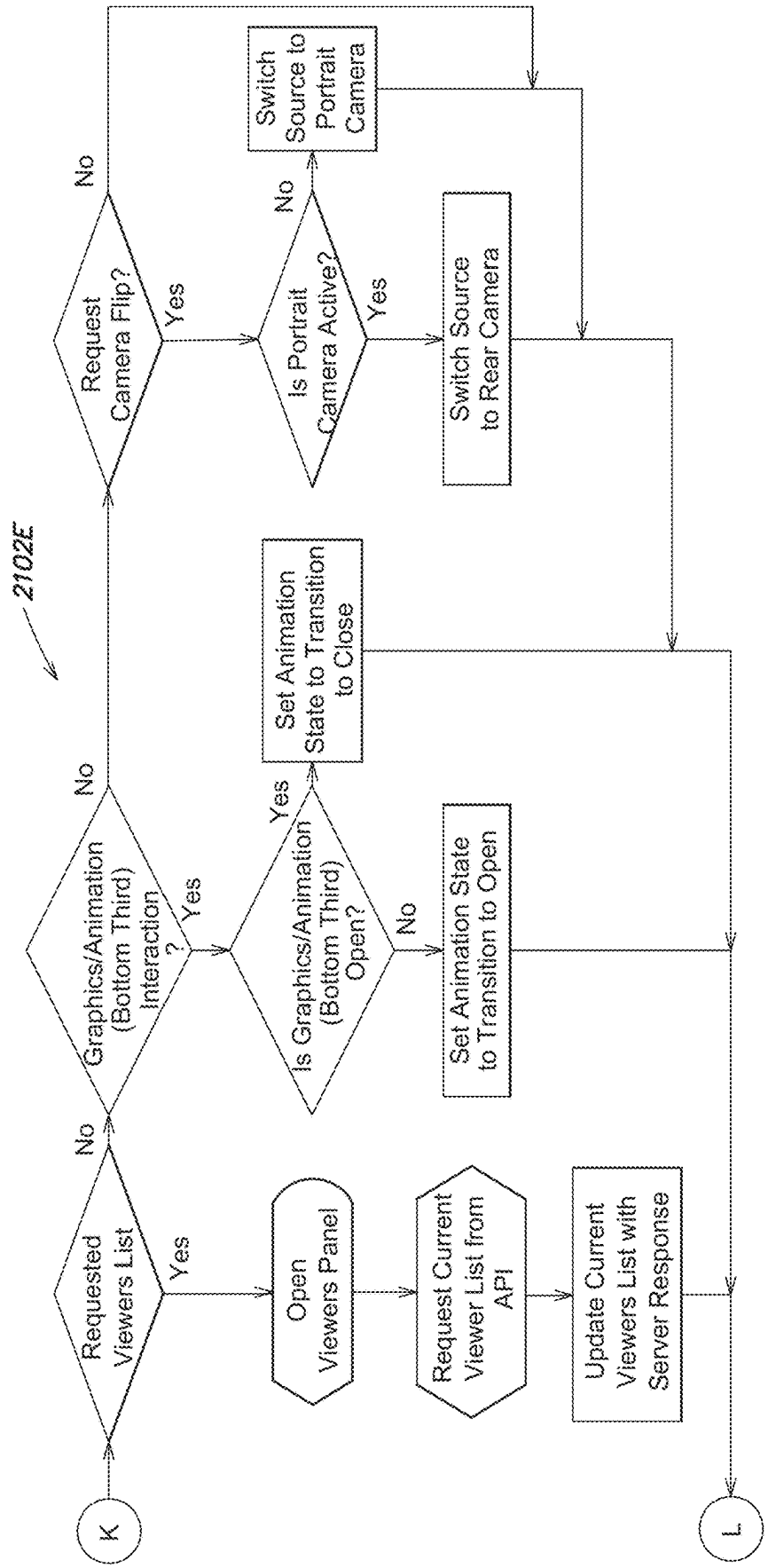

2102E
K
Requested Viewers List
No
Yes
Open Viewers Panel
Request Current Viewer List from API
Update Current Viewers List with Server Response
L
Graphics/Animation (Bottom Third) Interaction ?
No
Yes
Is Graphics/Animation (Bottom Third) Open?
Yes
No
Set Animation State to Transition to Close
Set Animation State to Transition to Open
Request Camera Flip?
No
Yes
Is Portrait Camera Active?
No
Yes
Switch Source to Portrait Camera
Switch Source to Rear Camera

FIG. 21E

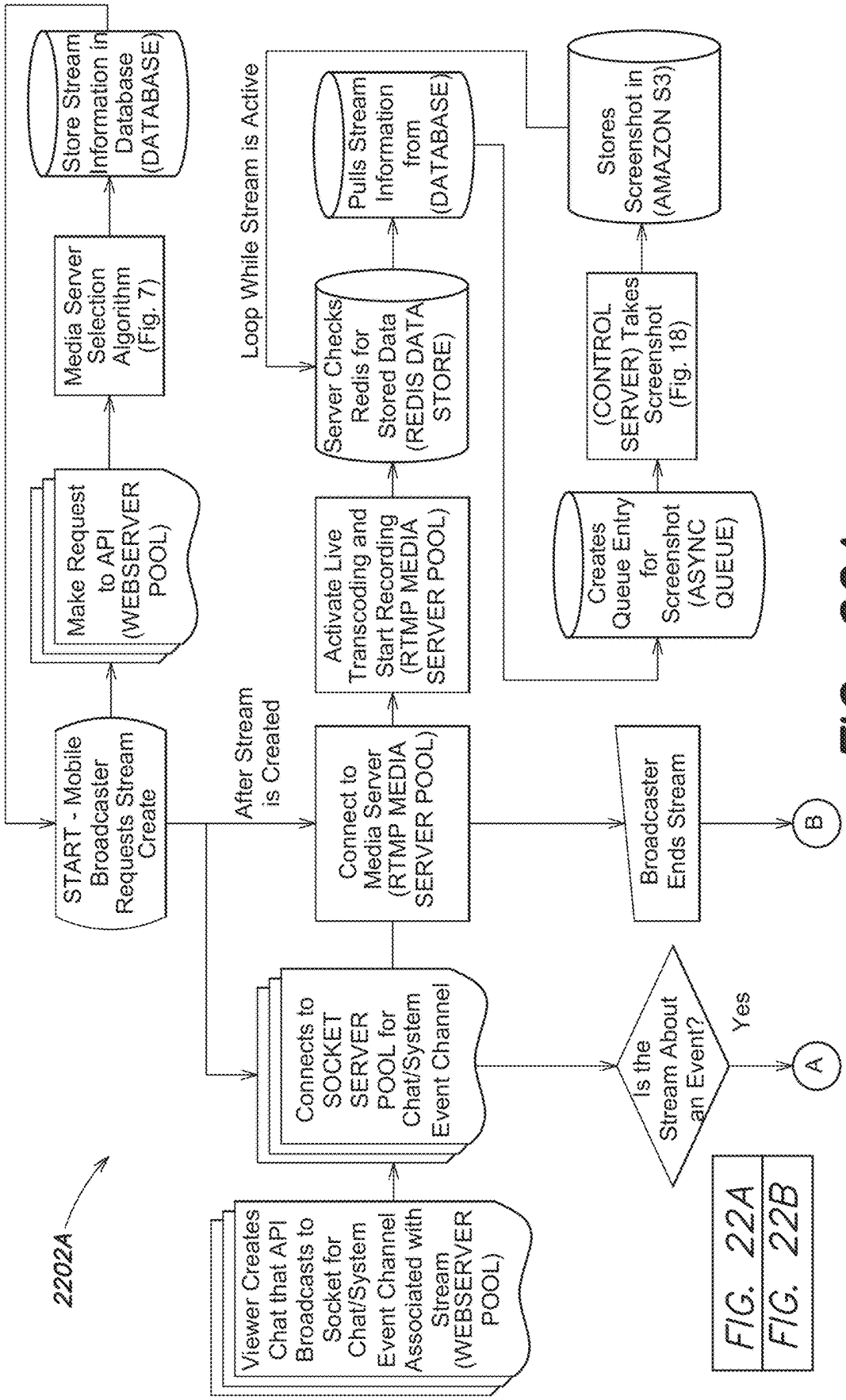


FIG. 22A

FIG. 22A
FIG. 22B

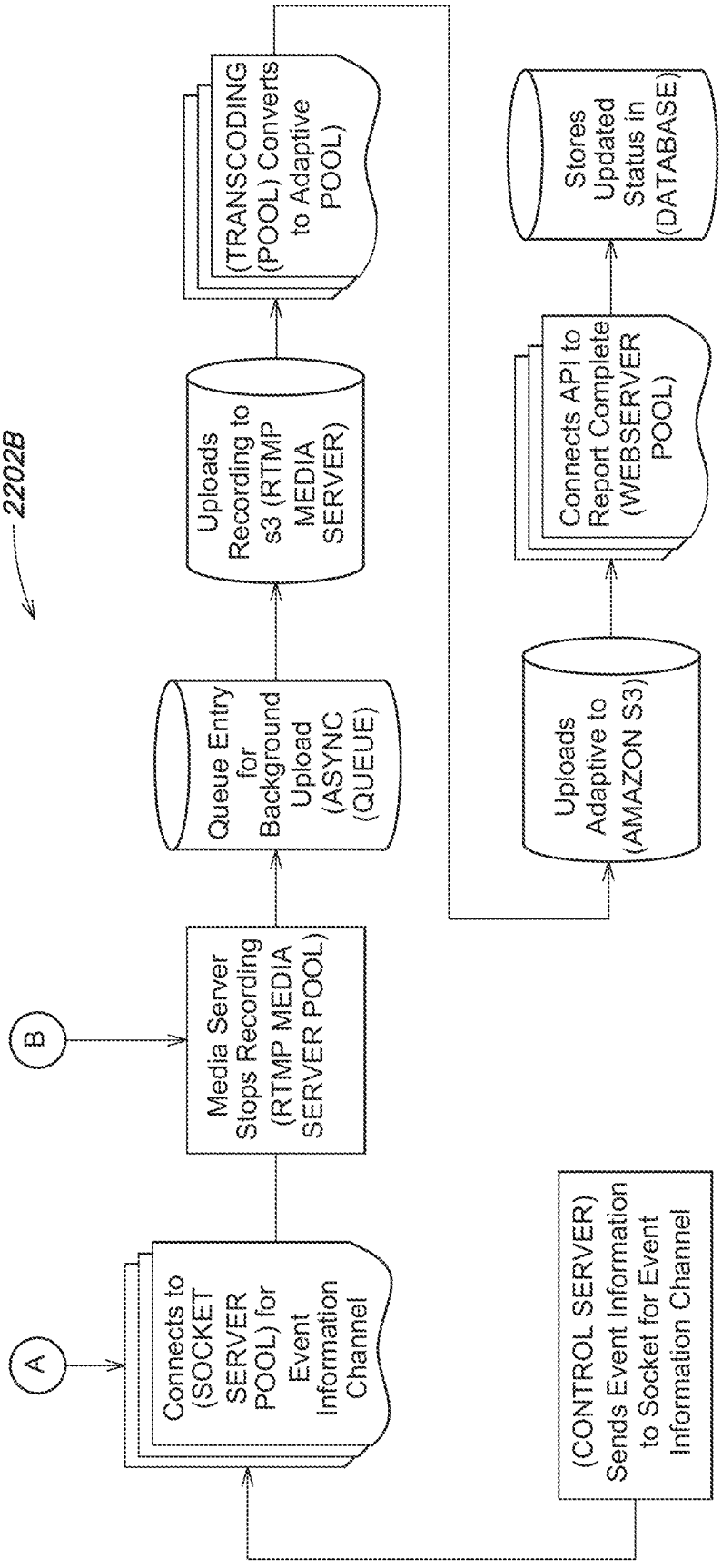

2202B
A
B
Connects to (SOCKET SERVER POOL) for Event Information Channel
Media Server Stops Recording (RTMP MEDIA SERVER POOL)
(CONTROL SERVER) Sends Event Information to Socket for Event Information Channel
Queue Entry for Background Upload (ASYNC (QUEUE)
Uploads Recording to s3 (RTMP MEDIA SERVER)
(TRANSCODING (POOL) Converts to Adaptive POOL)
Uploads Adaptive to (AMAZON S3)
Connects API to Report Complete (WEBSERVER POOL)
Stores Updated Status in (DATABASE)

FIG. 22B

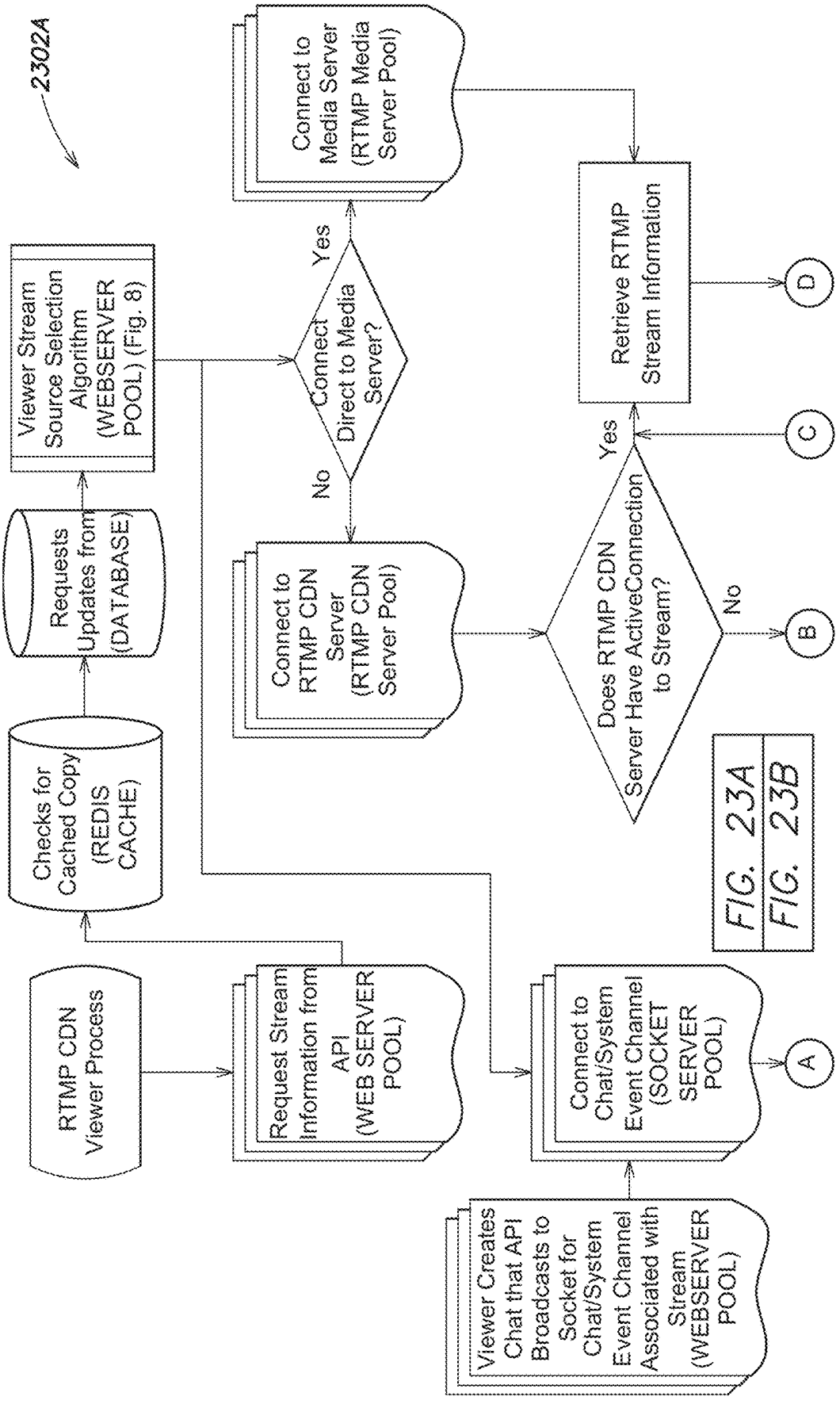


FIG. 23A
FIG. 23B

FIG. 23A

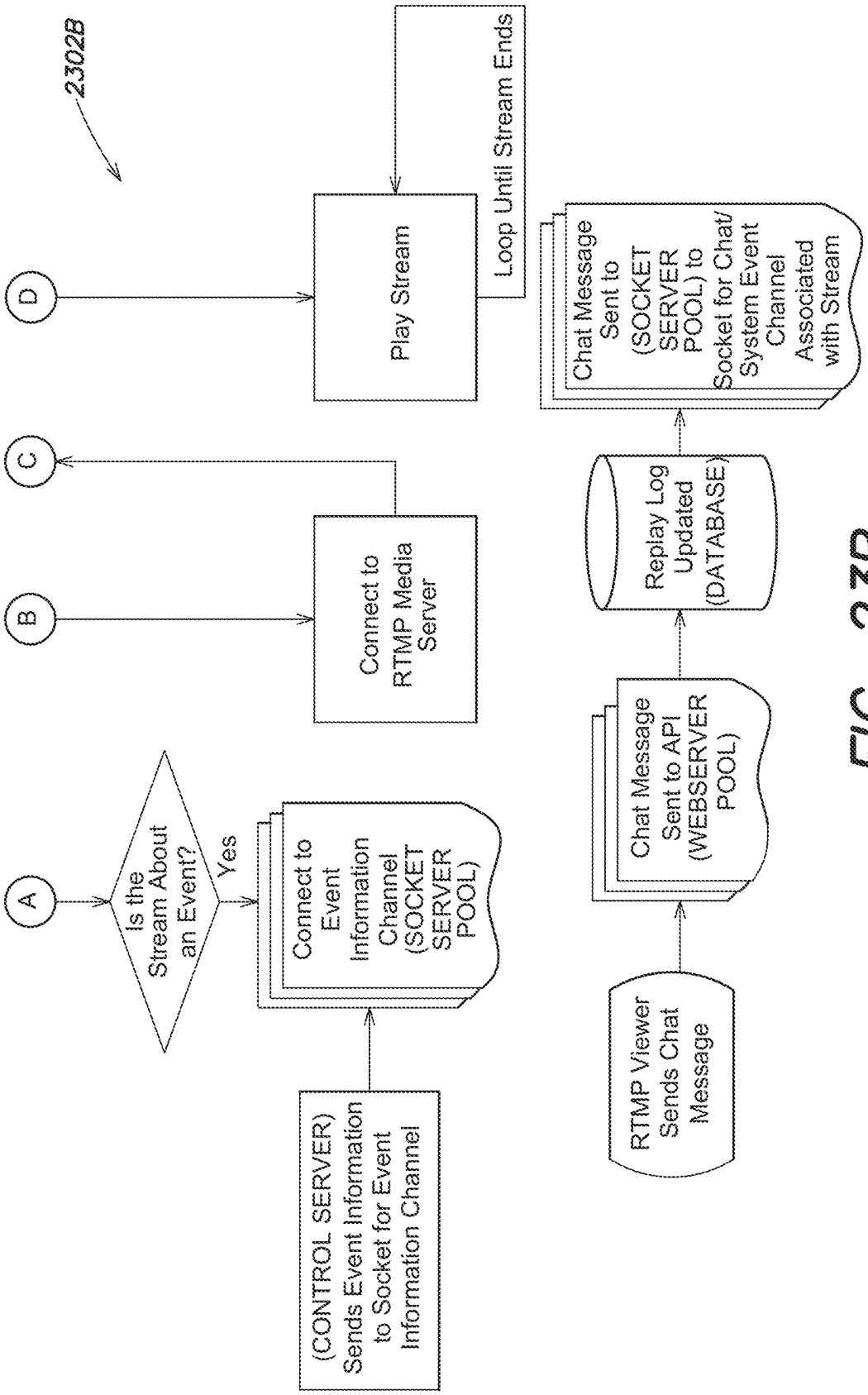

2302B
A
B
C
D
Is the Stream About an Event?
Yes
Connect to Event Information Channel (SOCKET SERVER POOL)
Connect to RTMP Media Server
Play Stream
Loop Until Stream Ends
(CONTROL SERVER) Sends Event Information to Socket for Event Information Channel
RTMP Viewer Sends Chat Message
Chat Message Sent to API (WEBSERVER POOL)
Replay Log Updated (DATABASE)
Chat Message Sent to (SOCKET SERVER POOL) to Socket for Chat/System Event Channel Associated with Stream

FIG. 23B

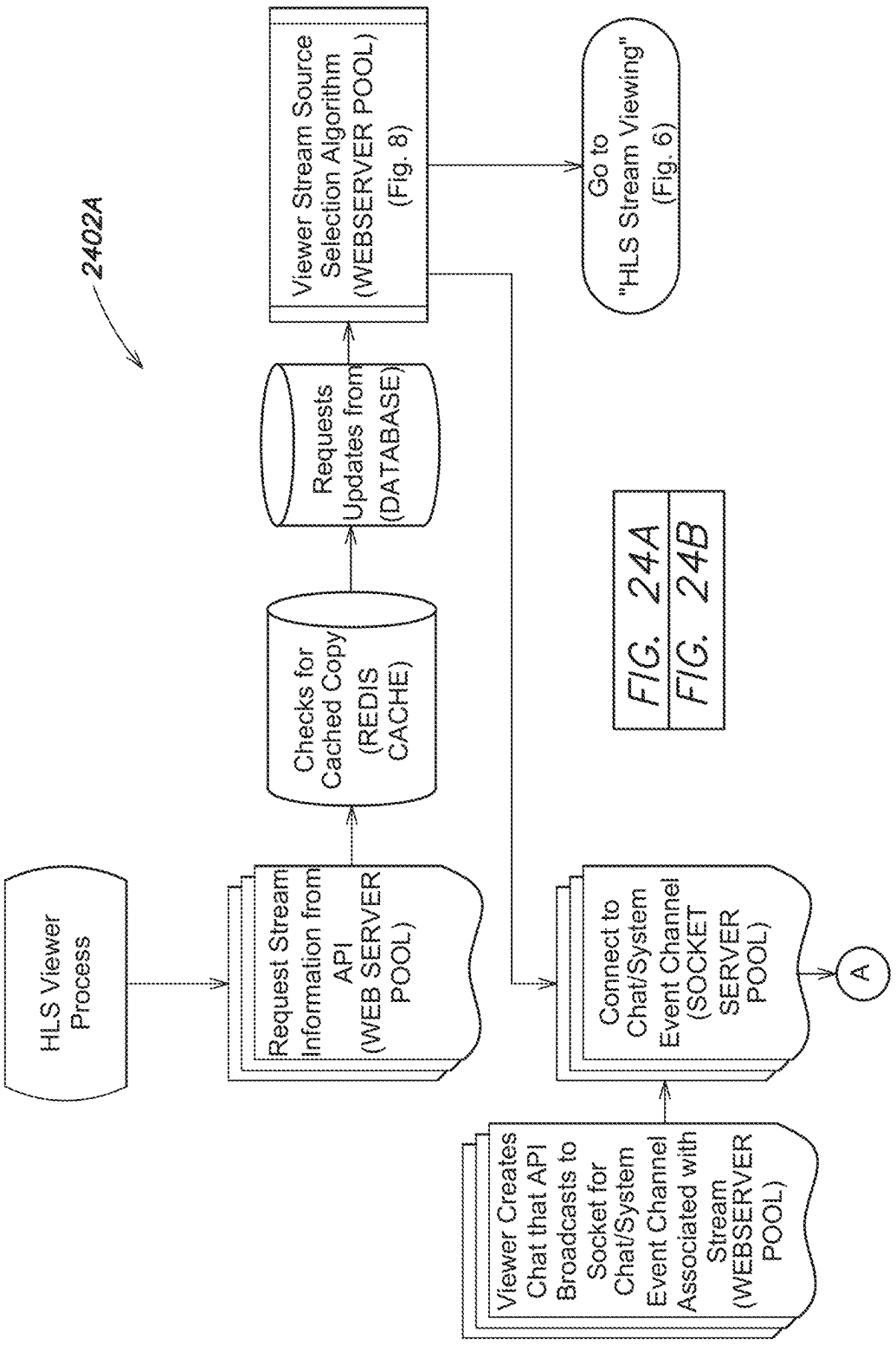

2402A
HLS Viewer Process
Request Stream Information from API (WEB SERVER POOL)
Checks for Cached Copy (REDIS CACHE)
Requests Updates from (DATABASE)
Viewer Stream Source Selection Algorithm (WEBSERVER POOL) (Fig. 8)
Go to "HLS Stream Viewing" (Fig. 6)
Connect to Chat/System Event Channel (SOCKET SERVER POOL)
A
Viewer Creates Chat that API Broadcasts to Socket for Chat/System Event Channel Associated with Stream (WEBSERVER POOL)
FIG. 24A
FIG. 24B

FIG. 24A

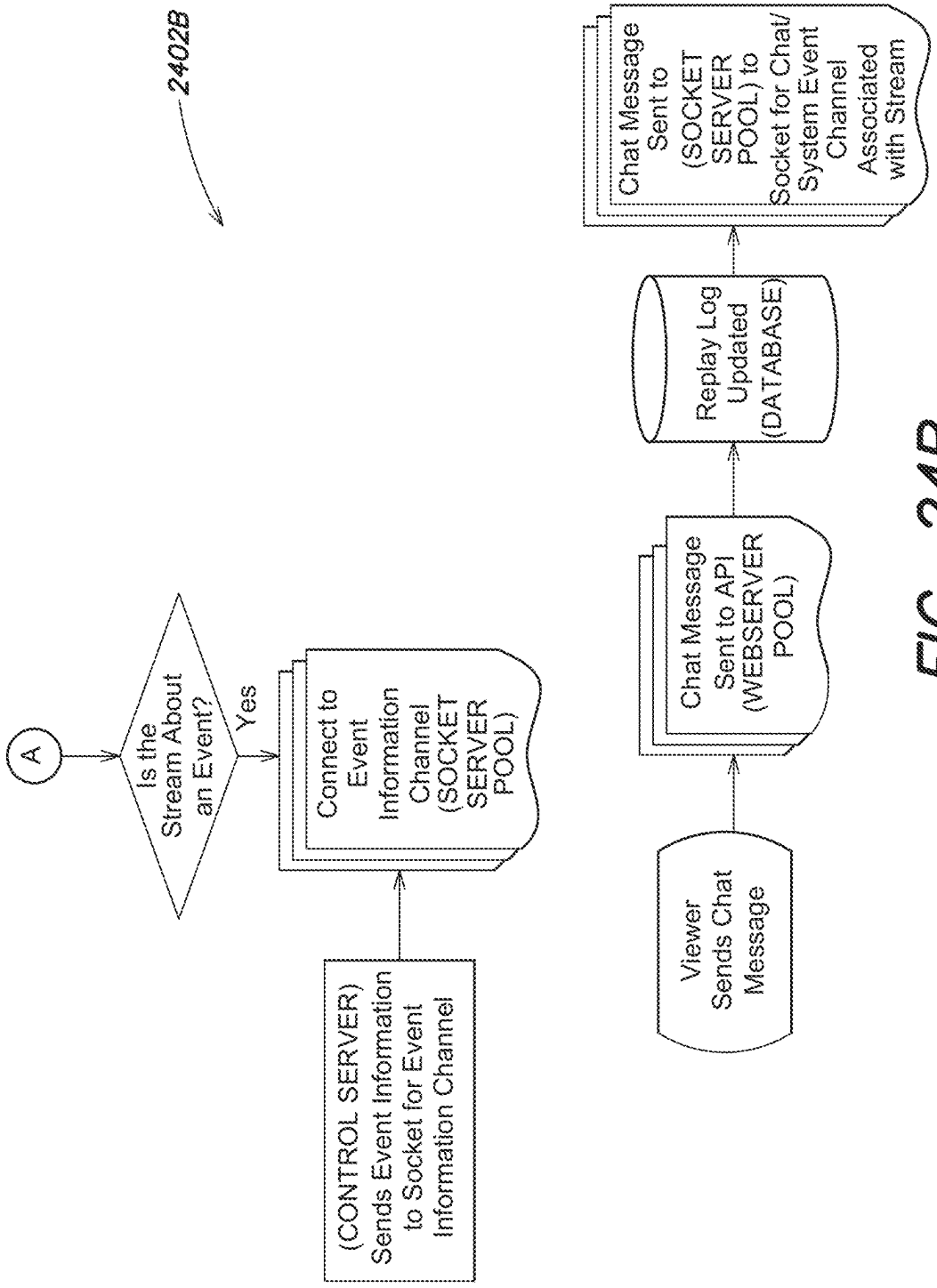

2402B
A
Is the Stream About an Event?
Yes
Connect to Event Information Channel (SOCKET SERVER POOL)
(CONTROL SERVER) Sends Event Information to Socket for Event Information Channel
Viewer Sends Chat Message
Chat Message Sent to API (WEBSERVER POOL)
Replay Log Updated (DATABASE)
Chat Message Sent to (SOCKET SERVER POOL) to Socket for Chat/System Event Channel Associated with Stream

FIG. 24B

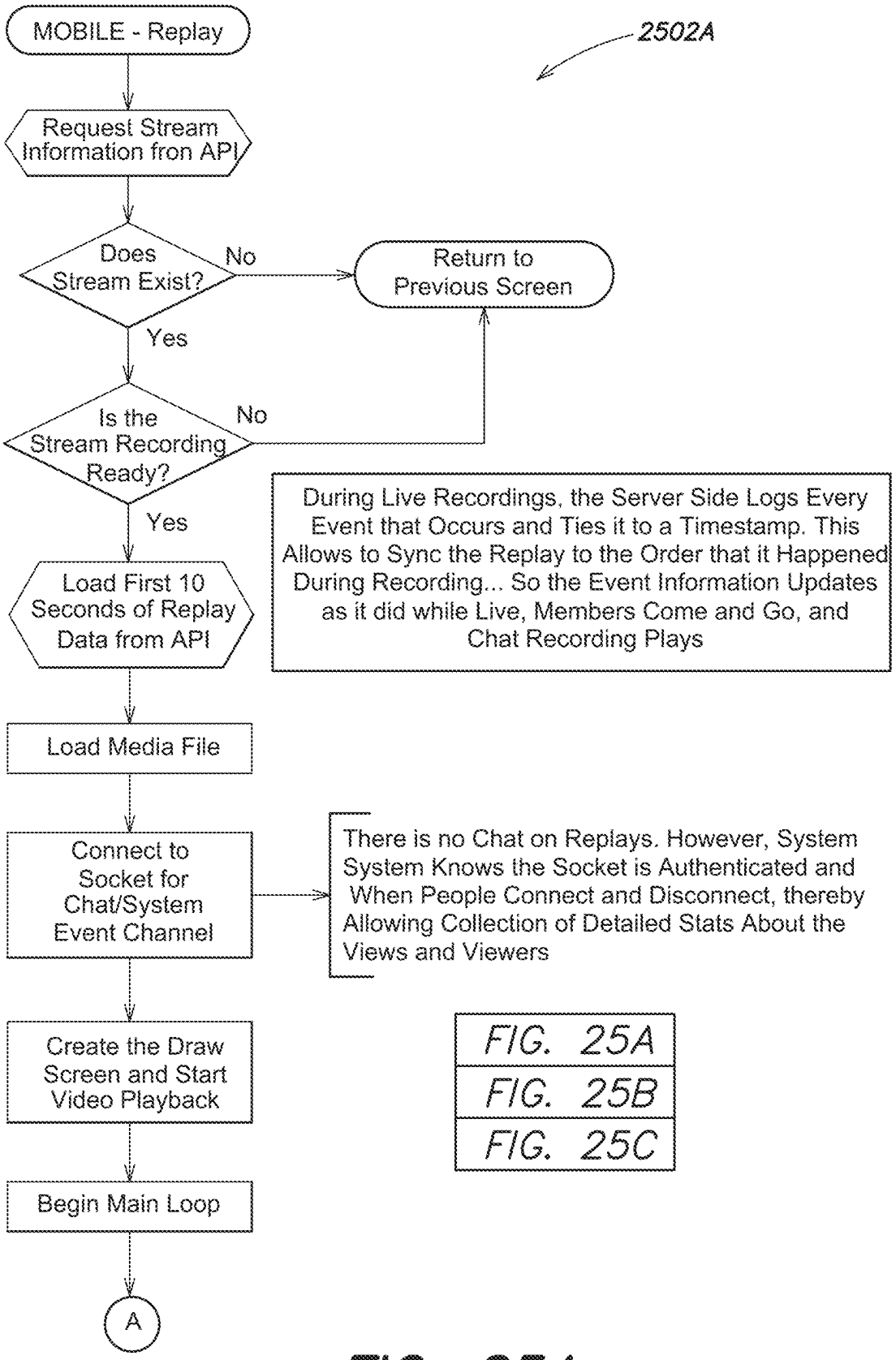


FIG. 25A
FIG. 25B
FIG. 25C

FIG. 25A

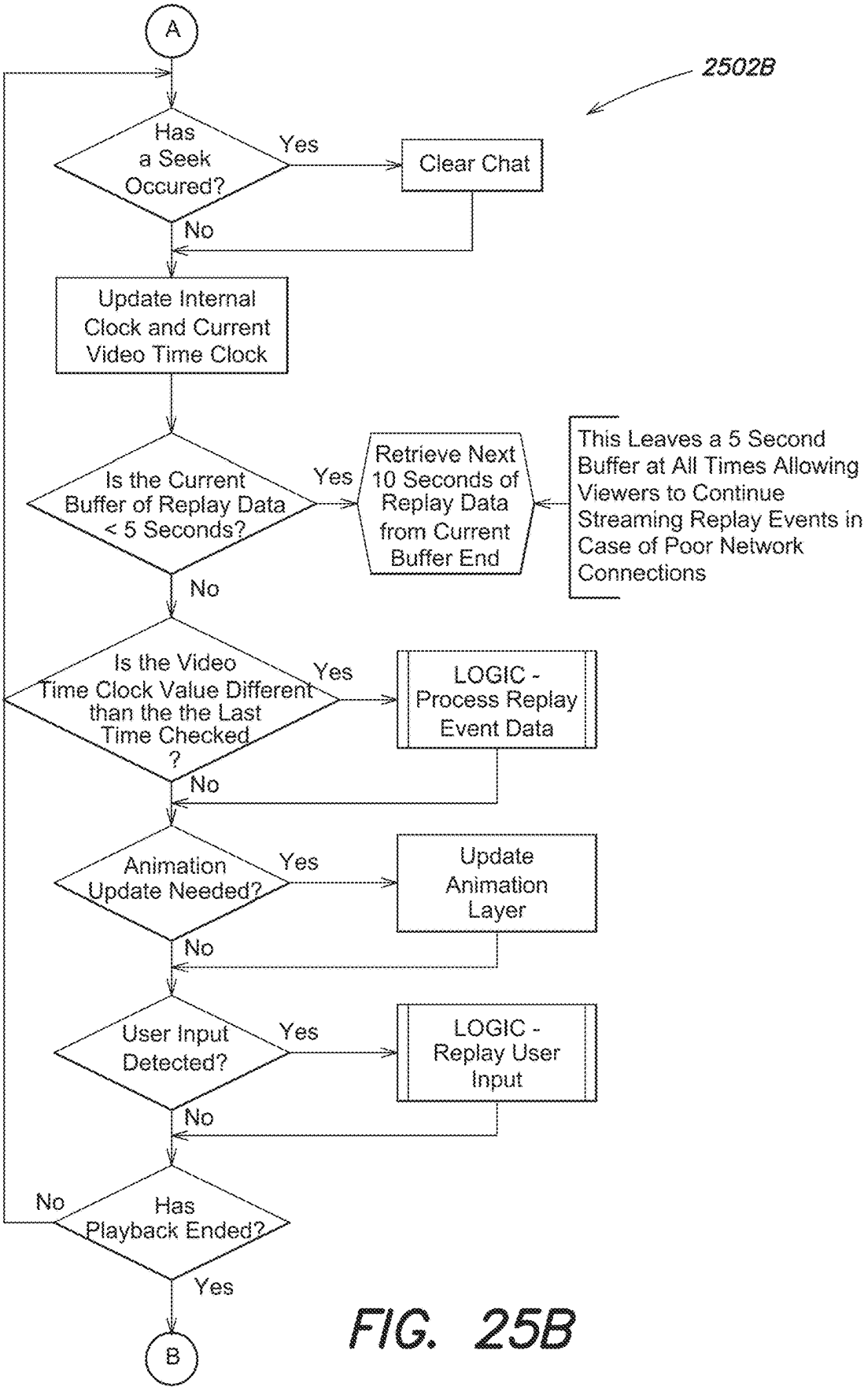

A
2502B
Has a Seek Occured?
Yes
Clear Chat
No
Update Internal Clock and Current Video Time Clock
Is the Current Buffer of Replay Data < 5 Seconds?
Yes
Retrieve Next 10 Seconds of Replay Data from Current Buffer End
This Leaves a 5 Second Buffer at All Times Allowing Viewers to Continue Streaming Replay Events in Case of Poor Network Connections
No
Is the Video Time Clock Value Different than the the Last Time Checked ?
Yes
LOGIC - Process Replay Event Data
No
Animation Update Needed?
Yes
Update Animation Layer
No
User Input Detected?
Yes
LOGIC - Replay User Input
No
Has Playback Ended?
No
Yes
B

FIG. 25B

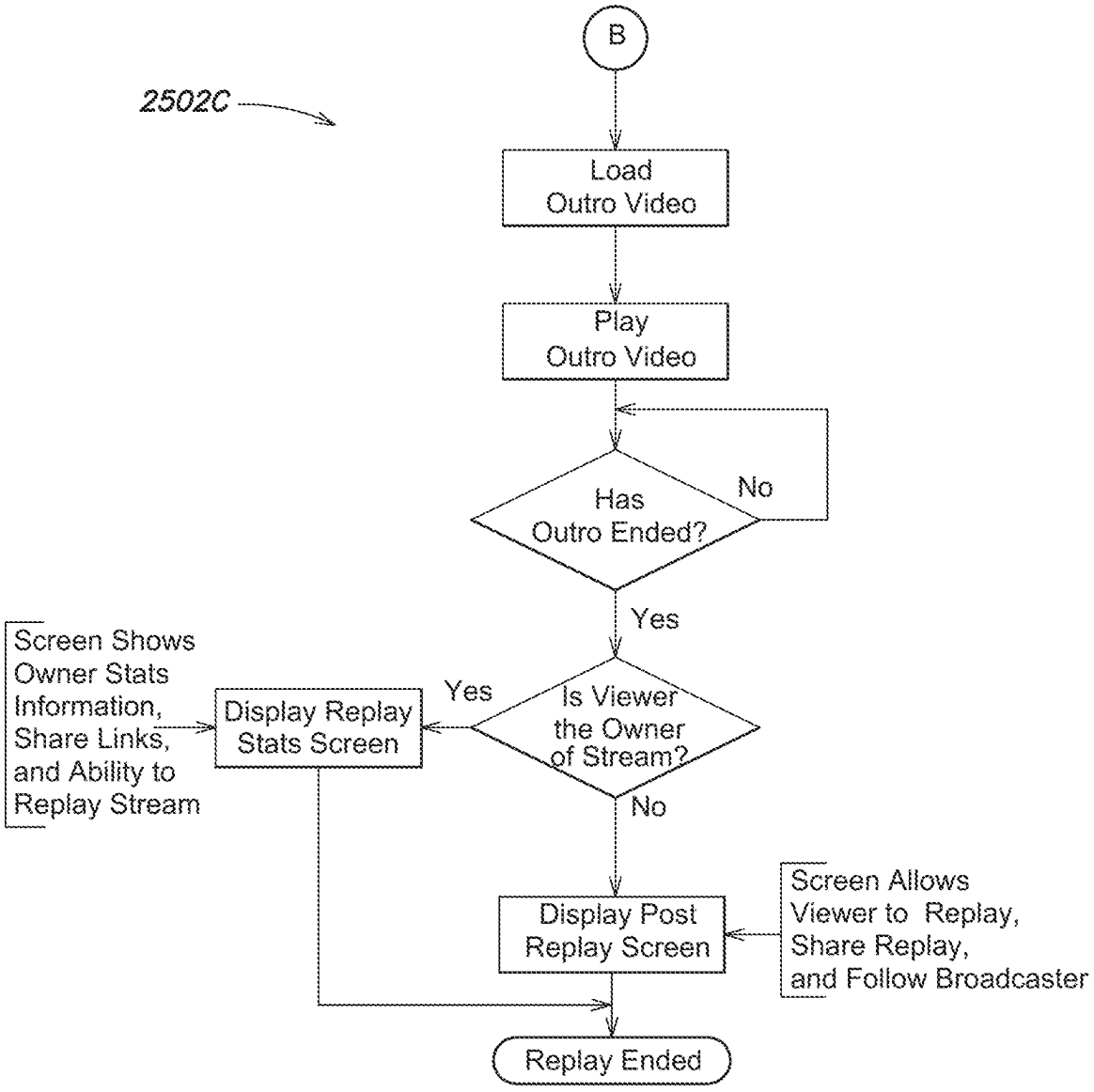

2502C
B
Load Outro Video
Play Outro Video
Has Outro Ended?
No
Yes
Is Viewer the Owner of Stream?
Yes
Display Replay Stats Screen
Screen Shows Owner Stats Information, Share Links, and Ability to Replay Stream
No
Display Post Replay Screen
Screen Allows Viewer to Replay, Share Replay, and Follow Broadcaster
Replay Ended

FIG. 25C

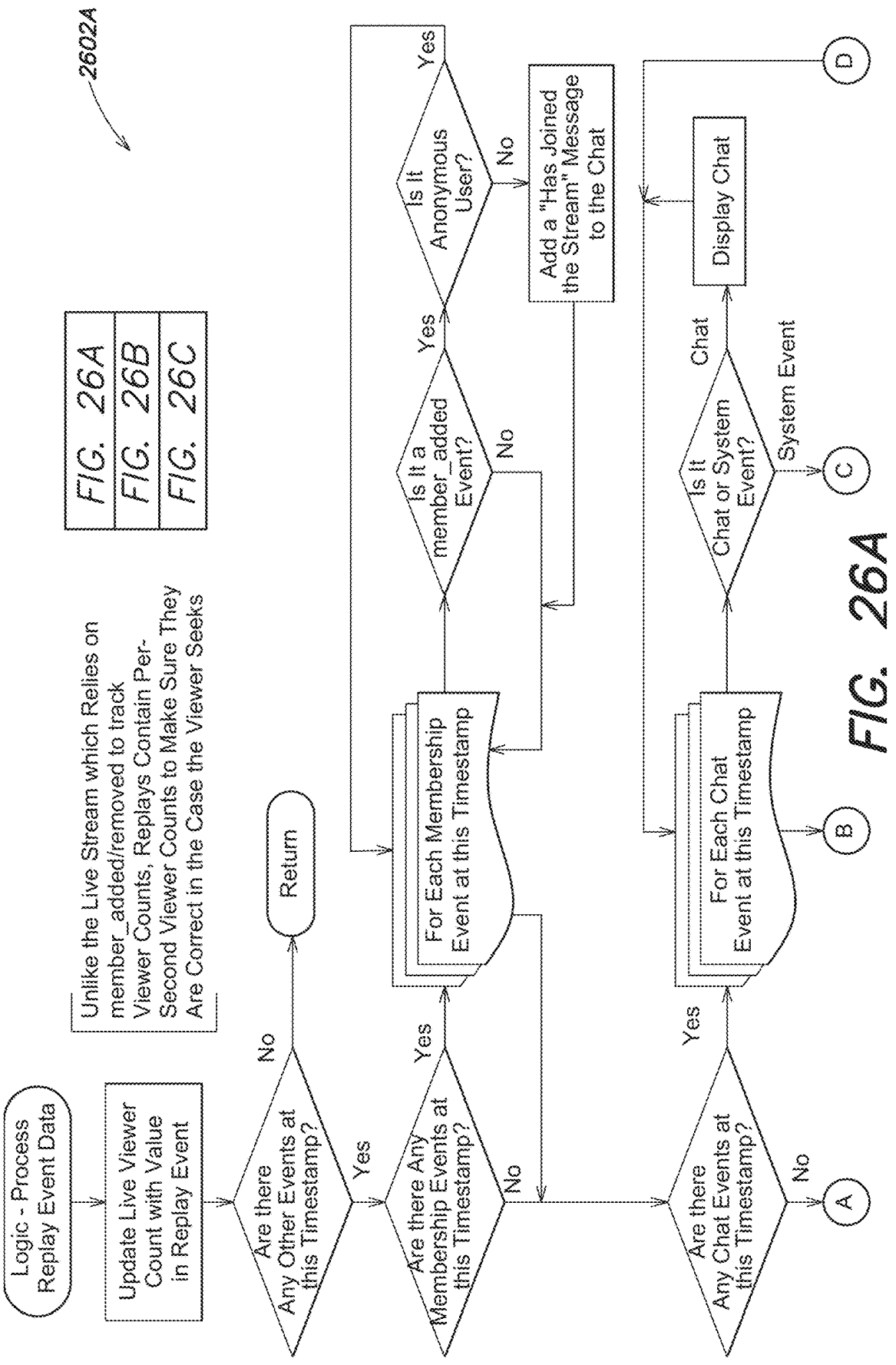


FIG. 26A
FIG. 26B
FIG. 26C

FIG. 26A

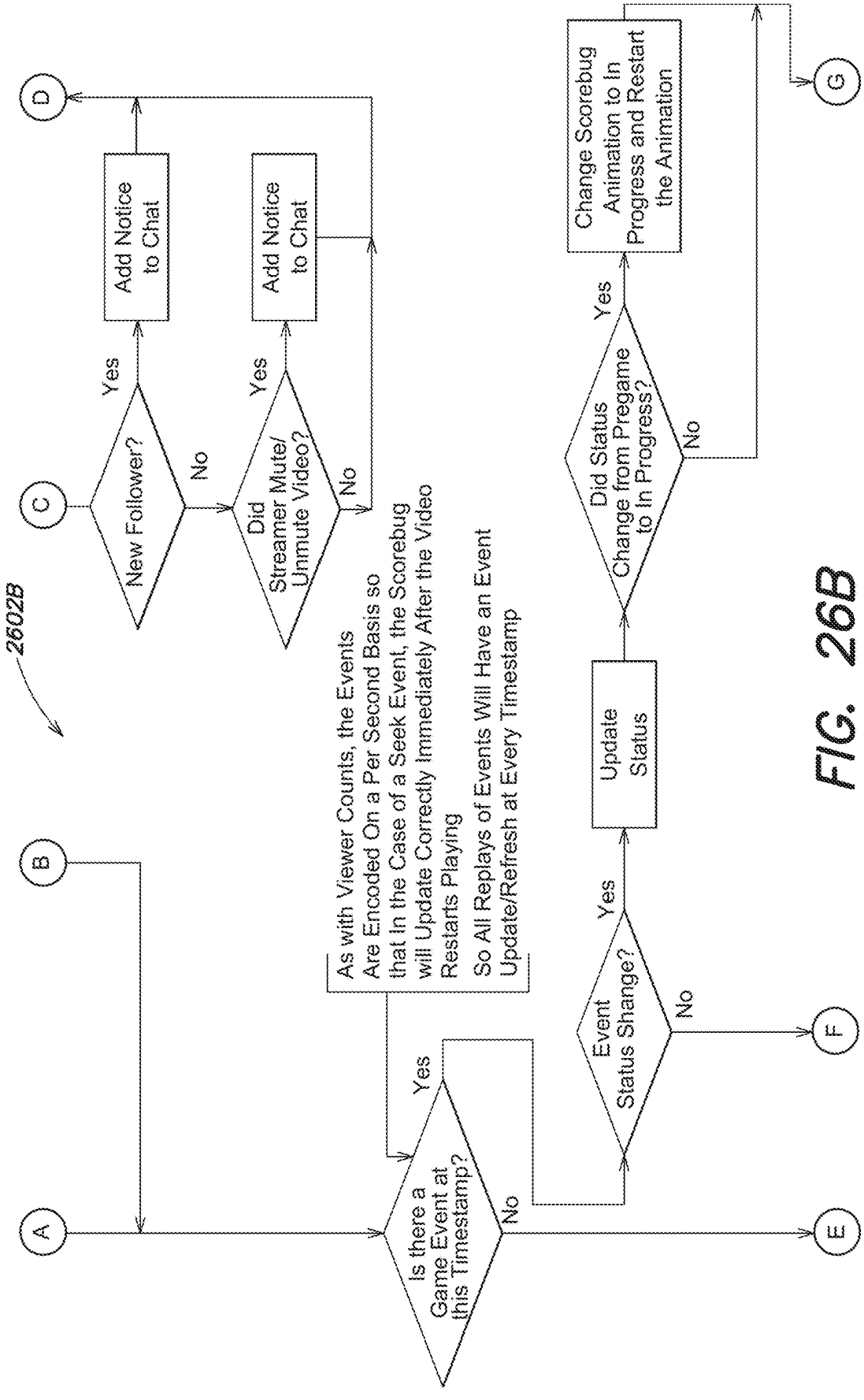

2602B
A
B
C
D
E
F
G
Is there a Game Event at this Timestamp?
Yes
No
As with Viewer Counts, the Events Are Encoded On a Per Second Basis so that In the Case of a Seek Event, the Scorebug will Update Correctly Immediately After the Video Restarts Playing
So All Replays of Events Will Have an Event Update/Refresh at Every Timestamp
Event Status Shange?
Yes
No
Update Status
Did Status Change from Pregame to In Progress?
Yes
No
Change Scorebug Animation to In Progress and Restart the Animation
New Follower?
Yes
No
Add Notice to Chat
Did Streamer Mute/ Unmute Video?
Yes
No
Add Notice to Chat

FIG. 26B

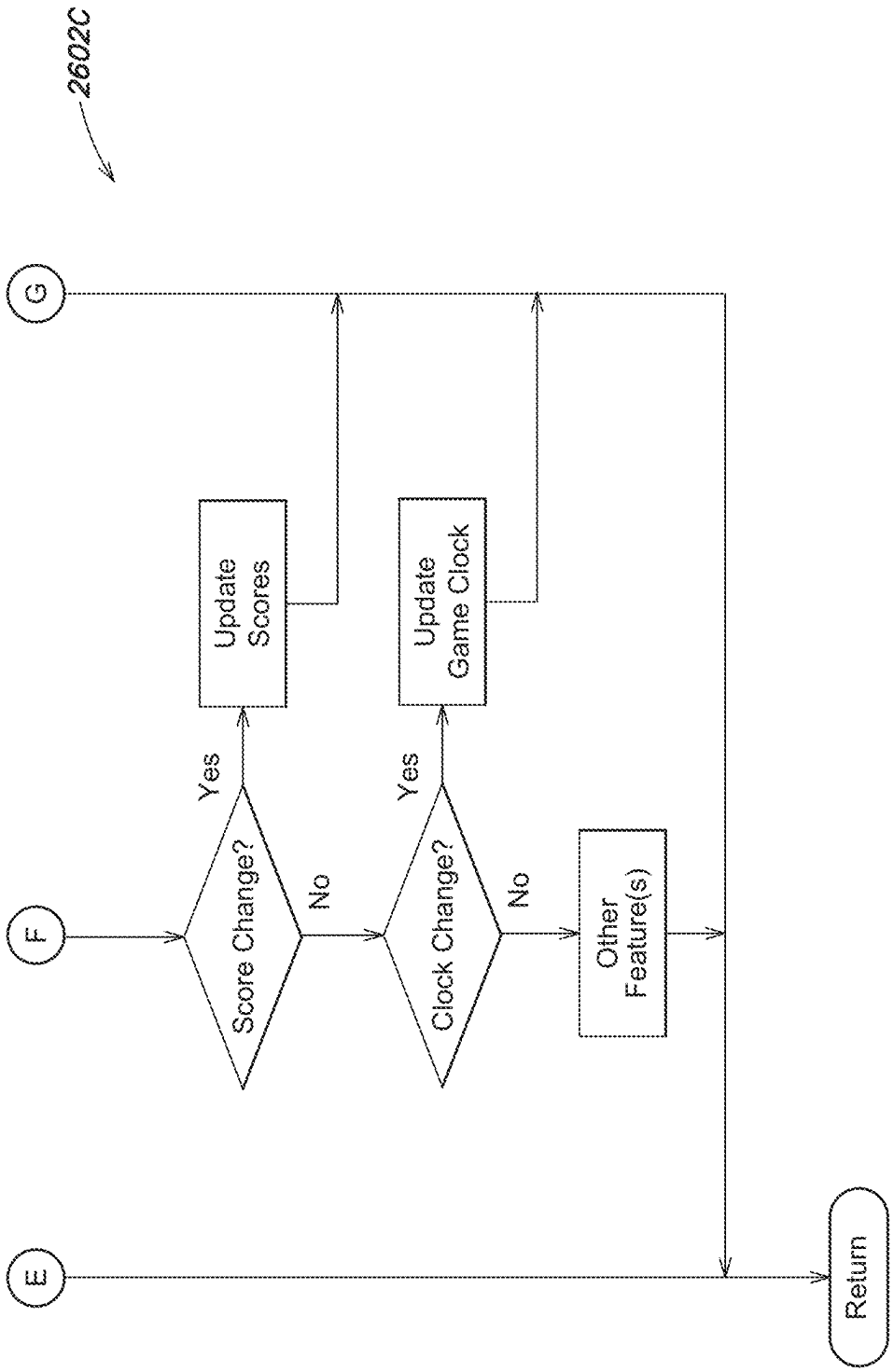

2602C
G
F
E
Score Change?
Yes
No
Update Scores
Clock Change?
Yes
No
Update Game Clock
Other Feature(s)
Return

FIG. 26C

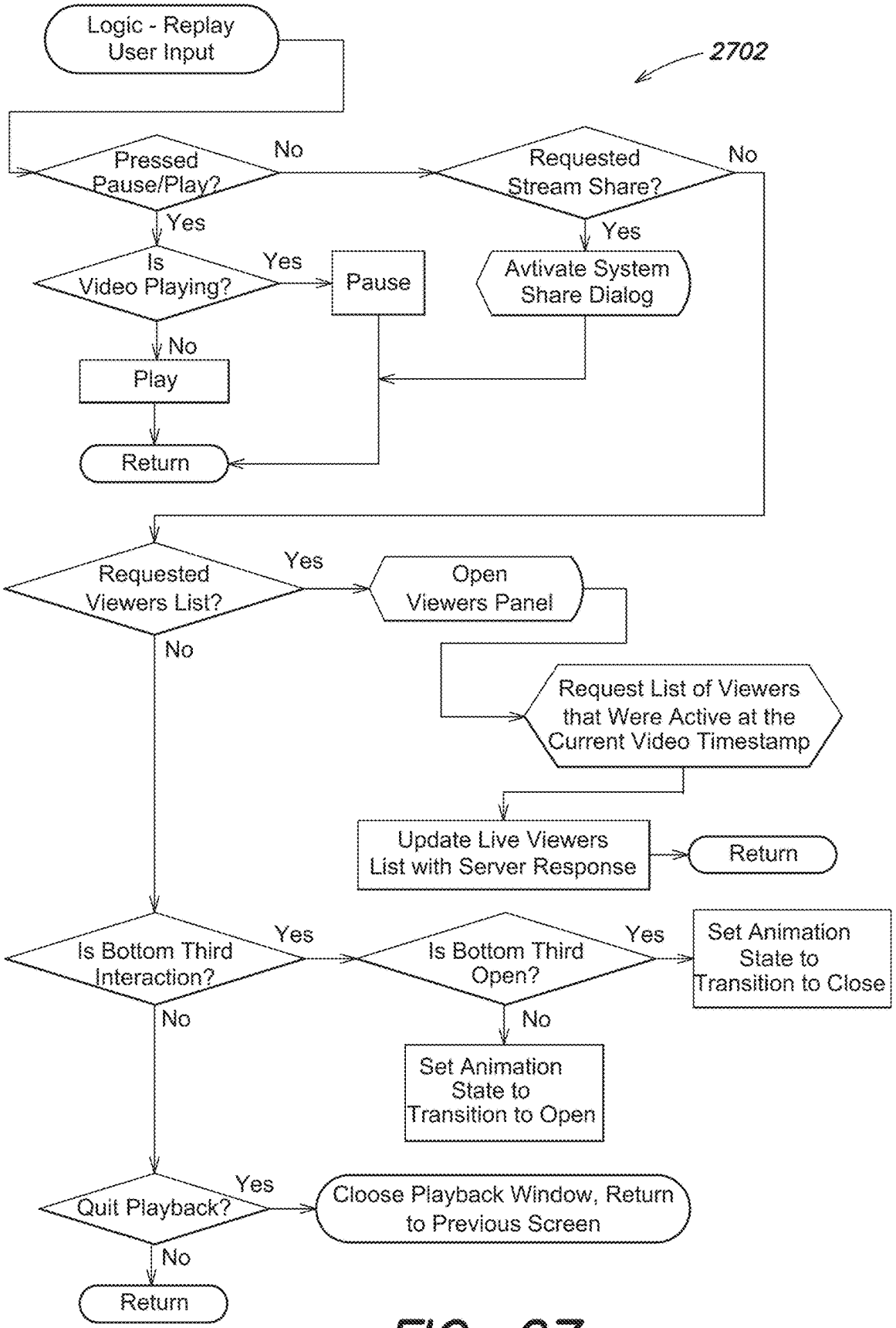

Logic - Replay User Input
2702
Pressed Pause/Play?
No
Requested Stream Share?
No
Yes
Yes
Is Video Playing?
Yes
Pause
Avtivate System Share Dialog
No
Play
Return
Requested Viewers List?
Yes
Open Viewers Panel
No
Request List of Viewers that Were Active at the Current Video Timestamp
Update Live Viewers List with Server Response
Return
Is Bottom Third Interaction?
Yes
Is Bottom Third Open?
Yes
Set Animation State to Transition to Close
No
No
Set Animation State to Transition to Open
Quit Playback?
Yes
Cloose Playback Window, Return to Previous Screen
No
Return

FIG. 27

SYSTEMS, APPARATUS, AND METHODS FOR RENDERING DIGITAL CONTENT STREAMS OF EVENTS, AND SYNCHRONIZATION OF EVENT INFORMATION WITH RENDERED STREAMS, VIA MULTIPLE INTERNET CHANNELS

CROSS-REFERENCES TO RELATED APPLICATIONS

The present application is a Continuation Application of U.S. application Ser. No. 17/027,219, filed Sep. 21, 2020, and entitled "SYSTEMS, APPARATUS, AND METHODS FOR RENDERING DIGITAL CONTENT STREAMS AND OVERLAYING RELATED INFORMATION WITH RENDERED STREAMS VIA MULTIPLE INTERNET CHANNELS," which is a Continuation Application of U.S. application Ser. No. 16/580,552, filed Sep. 24, 2019, and 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," which is a continuation of U.S. application Ser. No. 16/267,887, filed Feb. 5, 2019, 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," which is a Bypass Continuation Application of International Patent Application No. PCT/US2017/045801, filed Aug. 7, 2017, 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," which claims a priority benefit to the following U.S. provisional patent applications: U.S. Provisional Patent Application Ser. No. 62/371,558, filed Aug. 5, 2016, entitled "SYSTEMS, APPARATUS, AND METHODS FOR LIVE COMMENTARY;" U.S. Provisional Patent Application Ser. No. 62/435,361, filed Dec. 16, 2016, entitled "SYSTEMS, APPARATUS, AND METHODS FOR LIVE COMMENTARY;" U.S. Provisional Patent Application Ser. No. 62/485,878, filed Apr. 14, 2017, entitled "SYSTEMS, APPARATUS, AND METHODS FOR LIVE STREAMING, RECORDING, AND REPLAY OF BROADCAST VIDEO AND/OR AUDIO VIA THE INTERNET;" and U.S. Provisional Patent Application Ser. No. 62/518,506, filed Jun. 12, 2017, 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 AN HTTP LIVE STREAMING (HLS) CACHING SERVER ARCHITECTURE." Each of the above-identified applications is hereby incorporated herein by reference in its entirety.

BACKGROUND

The public's access to the Internet, as well as the amount of bandwidth for Internet users, increased greatly in the mid-1990s to the early 2000s. The available Internet bandwidth allowed for the streaming of both recorded audio and recorded video content to a user's home. Also, additional increase of Internet access was sparked by the rise of the smartphone and the ability for smartphones to access the Internet on-the-go. In 2005, YouTube™ began to offer Internet users the ability to upload, edit, view, rate, share and comment on a wide variety of user-generated and corporate media video content. Examples of such content include video clips, television show clips, music videos, audio recordings, movie trailers, video blogs, short original videos, and educational videos. Within one year of its inception, YouTube™ reached 65 thousand daily video uploads, with 100 million daily views.

The first Internet live video streaming platform in the United States, aptly called Livestream, was launched in 2007. Livestream, as well as other nascent live video streaming platforms, were content agnostic, and over time Internet viewers desired more specialized, content-specific, and niche live video streaming platforms. To accommodate evolving viewer demand, various live video streaming platforms have launched since Livestream; examples of such more specialized platforms include Twitch.tv (a live video streaming platform for creative arts and gaming content, which launched in 2011), and musical.ly (a live video streaming platform designed for music video content, which launched in 2014).

Live video streaming platforms have also aimed to attract social network users, and social networking platforms have evolved to include live video streaming capabilities. For example, Twitch.tv includes social networking components, and in 2015 the social media platform Twitter acquired the live video streaming platform Periscope. Other social media platforms have followed suit, with both Facebook and Instagram implementing live video streaming capabilities into their mobile applications in 2016. The widespread use of smartphones enables users of these social media platforms to share live videos with their social network.

SUMMARY

In spite of the proliferation of live video streaming over the Internet, the Inventors have recognized and appreciated various technological problems in conventional techniques for transmission of digital content via the Internet that adversely impact the live video streaming viewer experience. One such technological problem relates to 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. For example, 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. Other technological challenges for live video streaming arising from shortcomings in conventional techniques, as recognized and appreciated by the Inventors, include, for example, the 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), synchronizing such low latency and high quality copies of a live video stream amongst multiple viewers (particularly as the number of viewers of a given live video stream significantly increases), and 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).

With these various technological challenges in mind, the present disclosure relates generally to inventive systems, apparatus, and methods for facilitating one or more broadcasters to create/provide video and/or audio (also referred to herein as a "broadcast") and allow one or more viewers to consume the video and/or audio, either by receiving a copy of a live stream representing the video and/or audio essentially in real-time as created/provided by a given broadcaster, or by retrieving and playing a recording of the live stream at a later time. In the context of essentially real-time viewing of live streams, in various implementations the inventive systems, apparatus and methods discussed in detail herein address one or more technological problems relating to viewer latency, synchronization amongst different numbers of viewers, and providing scalable and flexible access to live streams to different classes/types of viewers and/or with different qualities of service.

With reference to FIG. **1**A, in various implementations a given broadcaster uses a network-connected client device (e.g., a first smart phone or other personal computing device communicatively coupled to the Internet) to generate a live stream of digital content corresponding to the video and/or audio created/provided by the broadcaster, wherein the live stream is transmitted to an inventive server and memory storage architecture (additional details of the server and memory storage architecture are shown, for example, in FIGS. **2** and **3**). The inventive server and memory storage architecture processes the broadcaster's live stream to generate multiple copies of the live stream which are provided to respective viewers, and also records the live stream and stores the recording for later replay. One or more viewers using different network-connected client devices (e.g., a second and third smart phone or other personal computing device communicatively coupled to the Internet) may communicatively couple to the server and memory storage architecture to receive a copy of the live stream of the digital content as a real-time or essentially real-time consumer of the broadcast created/provided by the broadcaster, or retrieve and play a recording of the live stream generated by the broadcaster client device (and recorded by the server and memory storage architecture). In some implementations, a given broadcaster themselves may view their own broadcast as a viewer on the same client device (e.g., by opening a separate tab on their client device and connecting to the server and memory storage architecture as both a broadcaster and a viewer). In one aspect, the inventive server and memory storage architecture allows any number of broadcasters to generate respective live streams of digital content, and any number of viewers to receive respective copies of a given broadcaster's live stream of digital content. In another aspect, the inventive server and memory storage architecture allows a given viewer of a first broadcaster's live stream to effectively switch in essentially real-time to view one or more other broadcasters' live streams.

In some implementations, the inventive systems, apparatus, and methods further facilitate a social platform in tandem with broadcasting video and/or audio to one or more viewers, in which a given broadcaster and their viewer(s) may communicate with one another (e.g., via chat messages) in essentially real-time during a broadcast. In one aspect, one or more chat messages, as well as various viewer-related information (e.g., name, surname, profile photo), may be displayed on respective client devices used by the broadcaster and the one or more viewers as the video and/or audio is rendered on the respective client devices.

In some implementations, the video and/or audio created/provided by one or more broadcasters is video-based commentary about a live event being observed by a given broadcaster. In various aspects, the broadcaster may take the role of a main commentator for the event, or a "color commentator" (e.g., analyst or summarizer) for the event; in a given role, the broadcaster may describe the event as it occurs (e.g., provide chronological or "play-by-play" updates to "call" the event), and/or may provide expert analysis, background or anecdotal information about the event and/or participants in the event, and/or relevant statistics regarding the event, participants in the event, or related events. FIG. **1**B illustrates a display of an example client device, in which a broadcaster is providing video-based commentary about a sports game, and in which chat messages from one or more viewers, event information about the sports game, and other graphics and/or animations are displayed together with the broadcaster's video-based commentary.

Examples of various activities constituting events according to the present disclosure include, but are not limited to, a variety of sports games (e.g., professional, semi-professional, intramural, community-oriented) or sporting activities (e.g., exercise-related activities, physical training activities, fishing, hunting), musical performances, theatrical performances, other artistic or entertainment-oriented activities, speeches or presentations, political activities (e.g., election-related activities, campaign-related activities, public or "town hall" meetings, public marches or demonstrations), military activities, professional activities (e.g., meetings or conferences), academic or educational activities (e.g., academic presentations or lectures, research activities, medical or scientific procedures, ceremonies), cooking or baking activities, competitive activities (e.g., racing activities), game-related activities (e.g., online gaming, board games, chess matches, role-playing games), social activities, and news-related activities. In one aspect, the video-based commentary provided by one or more broadcasters about the event and consumed by one or more viewers may provide a "second screen" experience for the viewers; in particular, in some implementations, the viewers may consume the video-based commentary about the event on their respective client devices as they are watching the event itself on another device (e.g., a television), or watching and/or participating in the event at the event's venue.

As discussed in greater detail below, in various aspects the inventive systems, apparatus and methods described herein provide an improvement in computer network functionality by facilitating scalable and appreciably low-latency viewing of copies of multiple broadcasters' live streams of video-based commentary about an event by significant numbers of viewers as the event unfolds. Particularly in the context of a "second screen" experience relating to a live event, discernible delay between the observation of the event itself and a given broadcaster's video-based commentary would significantly undermine viewer experience. Accordingly, the inventive systems, apparatus and methods described herein specifically address multiple technological problems in conventional techniques relating to transport of digital content via the Internet by providing multiple technological solutions to ensure a low-latency viewing experience for significant numbers of viewers.

In such implementations relating to video-based commentary about events, the inventive systems, apparatus and methods disclosed herein further may facilitate display, on respective client devices used by the broadcaster and the one or more viewers, of various event information in tandem with the video-based commentary rendered on the respective client devices. For example, in connection with sports

games, displayed event information may include, but is not 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. In some implementations, such event information about a sports game may be rendered in the display of a client device as a "scorebug" that may include for example, team abbreviations or logos, team scores, game status (e.g., period, quarter, inning), and actual or elapsed time for the game, with updates to one or more information elements in essentially real-time as the game progresses. It should be readily appreciated that for other types of events, a wide variety of information germane to the event 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. In another aspect, various event information or other information germane to a given broadcaster's video-based commentary may be rendered on a viewer client device in a "lower third" (also sometimes referred to as a "bottom third") of a display (e.g., as an overlay to the video-based commentary).

In view of the foregoing, in other aspects the inventive systems, apparatus and methods described herein provide additional improvements in computer network functionality by facilitating scalable and appreciably low-latency synchronization of displayed event information with multiple broadcasters' live streams of video-based commentary about an event as viewed by significant numbers of viewers as the event unfolds. Particularly in the context of a "second screen" experience relating to a live event, discernible delay between the observation of the event itself and the status of event information displayed in tandem with a given broadcaster's video-based commentary would significantly undermine viewer experience. Accordingly, the inventive systems, apparatus and methods described herein specifically address multiple technological problems in conventional techniques relating to transport of digital content via the Internet by providing multiple technological solutions to ensure not only a low-latency viewing experience for significant numbers of viewers (e.g., relative to the event about which a broadcaster is providing video-based commentary), but appropriate synchronization of event information across all broadcasters of an event and their associated viewers.

In some implementations as discussed herein, these technological solutions contemplate multiple Internet communication channels to provide relevant and synchronized information to client devices (as would be readily appreciated in the relevant arts, a "communication channel" refers to a physical or logical connection over a transmission medium to convey information signals from one or more senders to one or more receivers). For example, in one implementation, a first "video" Internet communication channel (e.g., between a media server or other media source and a client device) conveys the digital content corresponding to the video-based commentary provided by the broadcaster and consumed by one or more viewers, and a second "event information" Internet communication channel (e.g., between a particular socket of a socket server and the client device) conveys the event information. In other implementations, an additional "chat/system event" Internet communication channel (e.g., between another socket of a socket server and the client device) is employed to convey chat information and other information relating to system events regarding a particular broadcaster's live stream. In one aspect, connections between client devices and a particular socket of a socket server are persistent authenticated connections, so that the number of users (broadcasters and viewers) connected to a particular socket (e.g., and currently watching a particular live stream and/or particular event) may be tracked.

In some implementations, the inventive systems, apparatus, and methods further facilitate a variety of screen animations (e.g., motion graphics), customized displays or screen backgrounds, and other special effects graphics (e.g., augmented reality) that are generally associated with the video and/or audio created by a broadcaster, and rendered by client devices in tandem with a given broadcaster's video and/or audio.

In some implementations in which the broadcaster may be creating/providing video and/or audio content about an event, one or more such screen animations, customized displays or screen backgrounds, and other special effects graphics may be related to some aspect of the event. For example, as noted above, a "scorebug" providing information about a sporting event may be presented on a viewer's display with a variety of motion or other special effects graphics. In one aspect, the information provided by such a scorebug is derived from the synchronized event information that is received by the client device on an event information Internet communication channel that is separate from the video content representing the video-based commentary. In another aspect, the scorebug and/or other special effects graphics or animations may be rendered in the display of the client device pursuant to execution of a client app or program on the client device (rather than having these display elements integrated into the broadcaster's live stream itself). In other aspects, one or more such screen animations and other special effects may be provided in a lower/bottom third of a client device's display, and may be interactive (e.g., in that a user of a client device may select or thumb-over an animation, special effect and/or scorebug to retrieve additional information and/or initiate other animations/graphics) and/or user-customizable (e.g., a broadcaster may select from one of multiple background displays so that they may appear to be in any of a variety of locations or venues, customize their video-based commentary with a broadcaster-generated lower third, and/or add create/provide introduction videos to be shown before their live stream begins).

In yet other aspects, screen animations, customized displays or screen backgrounds, and/or other special effects graphics may be related to one or more of advertising or branding (e.g., by sponsors of a broadcaster, sponsors of an event about which a broadcaster is providing commentary, participants in the event itself, etc.), premium features and digital gifts (e.g., provided by one or more viewers to one or more broadcasters or other viewers). For example, in some implementations, via a client app or software program executing on a client device, users will be able to purchase digital gifts for other users (e.g., a viewer following a particular broadcaster may purchase digital beers, various digital concession stand items, ticker tapes, penalty flags, etc.), and the digital gifts will appear in the broadcaster's profile and/or be represented on a display (e.g., as an icon or sprite) together with the broadcaster's video-based commentary. In one implementation, information regarding digital gifts may be communicated between client devices and the server and memory storage architecture via a chat/system

event Internet communication channel associated with a given broadcaster's live stream of digital content.

In some implementations, the video and/or audio created/provided by one or more broadcasters may relate to various types of news, other types of current or past events, and various topics of interest about which a given broadcaster wishes to provide commentary. For example, a given broadcaster, or multiple broadcasters (which may be globally-distributed broadcasters, e.g., a first broadcaster in Europe, a second broadcaster in Africa, and a third broadcaster in Asia) may wish to broadcast about an ongoing, recent or past event (e.g., news about a bombing, a fire, an arrest, an economic development, a political or military development, or any of the other activities discussed above) or a particular topic of interest (e.g., healthy eating or dieting, gardening, religion, dating, politics, culture, art, music, playing a musical instrument, learning a language, auto repair, real estate, business, economics, legal issues, global warming, space exploration, a particular TV program or series, a particular entertainment or sports personality, video games, hobbies, etc.). As noted above, various event information regarding the ongoing, recent or past event (or particular topic of interest) may be provided to respective viewer devices, in tandem with the video-based commentary, in a variety of form factors and in a manner that ensures appropriate synchronization of event information across all broadcasters of an event and their associated viewers.

Thus, in various examples discussed in greater detail below, systems, apparatus, and methods are disclosed for obtaining, substantially in real time, via at least one communication interface over at least one network, an audio and/or visual feed generated by a broadcaster client device during at least one live event, wherein the audio and/or visual feed includes commentary associated with the at least one live event, and a separate event data feed including real time information associated with the at least one live event. The commentary and the real time information associated with the at least one live event are transmitted via at least one processor communicatively coupled to the at least one communication interface, so as to provide, substantially in real time, streaming content via the at least one communication interface over the at least one network to at least one display device for concurrent display of the commentary and a graphical ticker based on the real time information associated with the event. The disclosed systems, apparatus, and methods are enhanced tools for real-time broadcasting of user-generated content associated with one or more live events and community engagement that are unavailable from existing video hosting websites.

Glossary

USER: a person who interfaces, via a mobile app or web portal accessed on a client device communicatively coupled to the Internet, with one or more of the various servers and corresponding server functionality described herein.

REGISTERED USER: A user that provides profile information and validation credentials to establish a user account so as to access, via a login process using the validation credentials, the one or more of the various servers and corresponding server functionality described herein.

ANONYMOUS USER: A non-registered user that has access, without requiring a login process, to a limited feature set based on the server functionality described herein.

BROADCASTER: A registered user that creates/provides video and/or audio (also referred to herein in some instances as "video-based commentary") for consumption by one or more viewers.

EVENT: An activity about which a broadcaster may create/provide video and/or audio as the activity occurs (i.e., in "real-time"). Examples of activities constituting events according to the present disclosure include, but are not limited to, a variety of sports games (e.g., professional, semi-professional, intramural, community-oriented) or sporting activities (e.g., exercise-related activities, physical training activities, fishing, hunting), musical performances, theatrical performances, other artistic or entertainment-oriented activities, speeches or presentations, political activities (e.g., election-related activities, campaign-related activities, public or "town hall" meetings, public marches or demonstrations), military activities, professional activities (e.g., meetings or conferences), academic or educational activities (e.g., academic presentations or lectures, research activities, medical or scientific procedures, ceremonies), cooking or baking activities, competitive activities (e.g., racing activities), game-related activities (e.g., online gaming, board games, chess matches, role playing games), social activities, and news-related activities. Video and/or audio created/provided by a broadcaster about an event may be referred to herein as "video-based commentary." In some implementations, the video and/or audio created/provided by one or more broadcasters may relate to various types of news, other types of current, recent or past events, and various topics of interest about which a given broadcaster wishes to provide commentary (e.g., news about a bombing, a fire, an arrest, an economic development, a political or military development, or any of the other activities discussed above, healthy eating or dieting, gardening, religion, dating, politics, culture, art, music, playing a musical instrument, learning a language, auto repair, real estate, business, economics, legal issues, global warming, space exploration, a particular TV program or series, a particular entertainment or sports personality, video games, hobbies, etc.).

VIEWER: A registered user or anonymous user that consumes video and/or audio created/provided by a broadcaster, via essentially real-time access to a live stream of digital content representing the video and/or audio, or via retrieval and playing of a recording of the live stream of digital content.

LIVE STREAM: Digital content (e.g., digital video and/or digital audio) that is transferred between at least two network-connected devices in real-time or essentially real-time as the corresponding video and/or audio codified as the digital content is created/provided by a broadcaster. Thus, a network-connected client device used by the broadcaster may generate a live stream of digital content corresponding to the video and/or audio created/provided by the broadcaster, and a viewer using a different network-connected client device may receive a copy of a live stream of the digital content as a real-time or essentially real-time consumer of the video and/or audio created/provided by the broadcaster. In some implementations, the video and/or audio created/provided by the broadcaster and represented in a live stream may be video-based commentary relating to an event. A live stream of digital content may have a variety of data formats (e.g., H.264 MPEG-4 Advanced Video Coding video compression; VP8 video compression) and transmission protocols, including persistent/continuous streaming transmission protocols (e.g., real time streaming protocol "RTSP;" real time messaging protocol "RTMP;" web real-time communication "WebRTC"), as well as segmented

and/or adaptive bitrate (ABR) protocols (e.g., Apple's HTTP Live Streaming "HLS;" Microsoft's HTTP Smooth Streaming "MSS;" Adobe's HTTP Dynamic Streaming "HDS;" standards-based ABR protocol "MPEG-DASH").

FOLLOWER: A registered user who is notified when a particular broadcaster is online.

SUBSCRIBER: A follower of a particular broadcaster that has paid for access to additional features and/or content not available to a follower (e.g., subscriber-specific chat channels).

VIP MEMBER: an admin-designated registered user that has additional access rights and high priority access to live stream media servers.

MEDIA MEMBER: an admin-designated registered user that is a media professional. In some implementations, the media member designation is used during some live streams to facilitate filtering and prioritization of comments/chat message for response by the broadcaster of the live stream.

ADMIN USER: a staff account flagged with administrative powers.

PRIVATE PROFILE: a profile for a user that has designated their content as private, only allowing direct link access to live/replay streams and archive access to broadcasters that the user has followed.

RAW VIDEO: A recording made by a media server of a live stream generated by a broadcaster. In one example implementation in which a broadcaster client device generates a video stream in H.264/AAC format and the media server is a Real Time Messaging Protocol (RTMP) media server, the raw video is produced by the RTMP media server by recording the broadcaster's live stream directly (to provide a 720p high definition feed or even higher definition feed). In another example in which a broadcaster client device generates a video stream in VP8/WebRTC format and the media server is a WebRTC media server, the WebRTC media server first transcodes the live stream from the broadcaster into a H.264 720p high definition (or even higher definition feed) and then records the transcoded feed to provide the raw video.

In sum, one inventive example is directed to a system (**1000**) for providing a first plurality of copies (**202**A, **202**B) of a first broadcaster's live stream of digital content (**102**A) including first live sporting event video-based commentary from a first broadcaster client device (**100**A) to a first plurality of viewer client devices (**200**A, **200**B), and for providing a second plurality of copies (**202**C, **202**D) of a second broadcaster's live stream of digital content (**102**B) including second live sporting event video-based commentary from a second broadcaster client device (**100**B) to a second plurality of viewer client devices (**200**C, **200**D). The system comprise: A) a plurality of media sources (**300**) to: receive the first broadcaster's live stream of digital content (**102**A) and the second broadcaster's live stream of digital content (**102**B); provide a first copy (**202**A) of the first plurality of copies to a first viewer client device (**200**A) of the first plurality of viewer client devices via a first video Internet communication channel (**204**A) between the plurality of media sources and the first viewer client device of the first plurality of viewer client devices; and provide a first copy (**202**C) of the second plurality of copies to a first viewer client device (**200**C) of the second plurality of viewer client devices via a second video Internet communication channel (**204**C) between the plurality of media sources and the first viewer client device of the second plurality of viewer client devices. The system further comprises: B) a control server (**500**) to periodically retrieve, via the Internet and from an event information provider (**55**), first event information (**502**A) germane to a first live sporting event, wherein the first event information includes first score information (**504**A) for the first live sporting event. The system further comprises: C) at least one socket server (**600**) communicatively coupled to the control server to: receive from the control server at least the first score information; and transmit the first score information (**504**A) to at least the first viewer client device (**200**A) of the first plurality of viewer client devices via a first event information Internet communication channel (**206**A) between at least one first event socket (**602**A) 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.

Another inventive example is directed to a system (**1000**) for providing a first plurality of copies (**202**A, **202**B) of a first broadcaster's live stream of digital content (**102**A) including first video-based commentary about a first live sporting event from a first broadcaster client device (**100**A) to a first plurality of viewer client devices (**200**A, **200**B), and for providing a second plurality of copies (**202**C, **202**D) of a second broadcaster's live stream of digital content (**102**B) including second video-based commentary about a second live sporting event from a second broadcaster client device (**100**B) to a second plurality of viewer client devices (**200**C, **200**D). The system comprises: A) a plurality of media sources (**300**) to: receive the first broadcaster's live stream of digital content (**102**A) and the second broadcaster's live stream of digital content (**102**B); provide a first copy (**202**A) of the first plurality of copies to a first viewer client device (**200**A) of the first plurality of viewer client devices via a first Internet communication channel (**204**A) between the plurality of media sources and the first viewer client device of the first plurality of viewer client devices; and provide a first copy (**202**C) of the second plurality of copies to a first viewer client device (**200**C) of the second plurality of viewer client devices via a second Internet communication channel (**204**C) between the plurality of media sources and the first viewer client device of the second plurality of viewer client devices. The system further comprises: B) a control server (**500**) to periodically retrieve, via the Internet and from an event information provider (**55**), first event information (**502**A) germane to the first live sporting event and second event information (**502**B) germane to the second live sporting event. The system further comprises: C) at least one socket server (**600**) communicatively coupled to the control server to: receive from the control server at least some of the first event information (**502**A) and at least some of the second event information (**502**B), wherein: the at least some of the first event information includes first score information for the first sporting event; and the at least some of the second event information includes second score information for the second sporting event; and transmit at least the first score information the at least some of the first event information (**502**A) to the first viewer client device (**200**A) of the first plurality of viewer client devices via a third Internet communication channel (**206**A) between at least one first event socket (**602**A) of the at least one socket server and the first viewer client device of the first plurality of viewer devices; and transmit at least the second score information of the at least some of the second event information (**502**B) to the first viewer client device (**200**C) of the second plurality of viewer client devices via a fourth Internet communication channel (**206**C) between at least one second

event socket (602B) of the at least one socket server and the first viewer client device of the second plurality of viewer devices.

It should be appreciated that all combinations of the foregoing concepts and additional concepts discussed in greater detail below (provided such concepts are not mutually inconsistent) are contemplated as being part of the inventive subject matter disclosed herein. In particular, all combinations of claimed subject matter appearing at the end of this disclosure are contemplated as being part of the inventive subject matter disclosed herein. It should also be appreciated that terminology explicitly employed herein that also may appear in any disclosure incorporated by reference should be accorded a meaning most consistent with the particular concepts disclosed herein.

Other systems, processes, and features will become apparent to those skilled in the art upon examination of the following drawings and detailed description. It is intended that all such additional systems, processes, and features be included within this description, be within the scope of the present invention, and be protected by the accompanying claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The skilled artisan will understand that the drawings primarily are for illustrative purposes and are not intended to limit the scope of the inventive subject matter described herein. The drawings are not necessarily to scale; in some instances, various aspects of the inventive subject matter disclosed herein may be shown exaggerated or enlarged in the drawings to facilitate an understanding of different features. In the drawings, like reference characters generally refer to like features (e.g., functionally similar and/or structurally similar elements).

FIG. 1A is a block diagram showing respective elements of a system to facilitate live streaming of digital content (video and/or audio) from multiple broadcasters to multiple viewers, according to some inventive implementations.

FIG. 1B illustrates a display of an example client device in the system of FIG. 1A, showing various displayed content according to some inventive implementations.

FIG. 2 is a block diagram of the broadcast/viewing servers and memory storage devices shown in FIG. 1A according to some inventive implementations.

FIG. 3 is a block diagram showing additional details of the various interconnections between the respective components of the servers and memory storage devices shown in FIG. 2, according to some inventive implementations.

FIGS. 4A and 4B show a process flow diagram illustrating a broadcast media server selection algorithm according to one inventive implementation, which in some examples may be performed by one or more web servers shown in FIGS. 2 and 3.

FIGS. 5A through 5C show a process flow illustrating a media server process for the media servers shown in FIGS. 2 and 3, according to one inventive implementation.

FIG. 6 is a block diagram illustrating the selective coupling of an example viewer client device to one of the media sources of the servers and memory storage devices shown in FIGS. 2 and 3, according to some inventive implementations.

FIG. 7 is a process flow diagram illustrating a viewer stream source selection algorithm according to one inventive implementation, which in some examples may be performed by the one or more web servers shown in FIGS. 2, 3 and 6.

FIG. 8 is a block diagram showing additional details of the HLS server architecture of the servers and memory storage devices shown in FIGS. 2, 3 and 6, according to some inventive implementations.

FIGS. 9A through 9D show a process flow illustrating an HLS stream viewing process performed by the HLS server architecture shown in FIG. 8, according to one inventive implementation.

FIG. 10 illustrates some of the functionality (e.g., services and other processes) performed by the control server shown in FIGS. 2 and 3, according to one inventive implementation.

FIGS. 11A through 11C show a process flow diagram illustrating an RTMP media server scaling system service method performed by the control server of FIG. 10, according to one inventive implementation.

FIGS. 12A through 12C show a process flow diagram illustrating an RTMP CDN server scaling system service method performed by the control server of FIG. 10, according to one inventive implementation.

FIGS. 13A and 13B show a process flow diagram illustrating a stream and server watchdog service method performed by the control server of FIG. 10, according to one inventive implementation.

FIGS. 14A and 14B show a process flow diagram illustrating a check RTNIP media/CDN server method performed by the control server of FIG. 10 (e.g., as part of the method shown in FIGS. 13A and 13B), according to one inventive implementation.

FIG. 15 shows a process flow diagram illustrating an event data ingress service method performed by the control server of FIG. 10, according to one inventive implementation.

FIGS. 16A and 16B show a process flow diagram illustrating a live event data monitor service method performed by the control server of FIG. 10, according to one inventive implementation.

FIGS. 17A and 17B show a process flow diagram illustrating an asynchronous task service method performed by the control server of FIG. 10, according to one inventive implementation.

FIGS. 18A and 18B show a process flow diagram illustrating a process for taking a screenshot (thumbnail) of a live stream, performed by the control server of FIG. 10, according to one inventive implementation.

FIGS. 19A and 19B show a process flow diagram illustrating a user login process according to one inventive implementation, which in some examples may be performed by a client device and facilitated by one or more web servers shown in FIGS. 2 and 3.

FIGS. 20A and 20B show a process flow diagram illustrating a mobile broadcaster stream create process according to one inventive implementation, which in some examples may be performed by a broadcaster client device shown in FIG. 1A and facilitated by one or more web servers shown in FIGS. 2 and 3.

FIGS. 21A, 21B, 21C, 21D, and 21E show a process flow illustrating a mobile broadcaster active stream process according to one inventive implementation, which in some examples may be performed at least in part by a broadcaster client device shown in FIG. 1A.

FIGS. 22A and 22B show a communication flow diagram illustrating process flow elements and the server and/or memory storage devices involved in the communication flow for the processes shown in FIGS. 20A and 20B, and FIGS. 21A-21E, according to one inventive implementation.

FIGS. **23**A and **23**B show a communication flow diagram illustrating process flow elements and the server and/or memory storage devices involved in the communication flow for a live stream RTMP media server or RTMP CDN viewer, according to one inventive implementation.

FIGS. **24**A and **24**B show a communication flow diagram illustrating process flow elements and the server and/or memory storage devices involved in the communication flow for a live stream HLS viewer, according to one inventive implementation.

FIGS. **25**A, **25**B, and **25**C show a process flow illustrating a mobile client live stream replay method, according to one inventive implementation.

FIGS. **26**A, **26**B, and **26**C show a process flow illustrating an event data replay process called in the method of FIGS. **25**A, **25**B and **25**C, according to one inventive implementation.

FIG. **27** shows a process flow illustrating a user input replay process called in the method of FIGS. **25**A, **25**B and **25**C, according to one inventive implementation.

DETAILED DESCRIPTION

Following below are more detailed descriptions of various concepts related to, and implementations of, inventive systems, methods and apparatus 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. It should be appreciated that various concepts introduced above and discussed in greater detail below may be implemented in various manners, and that examples of specific implementations and applications are provided primarily for illustrative purposes.

I. Overview

The present disclosure describes inventive systems, apparatus, and methods for connecting followers of live events (e.g., sports, performances, speeches, etc.), including commentators, spectators, and/or participants in live events (e.g., athletes, performers, politicians, etc.). In some example implementations, the inventive systems, apparatus and methods further provide a social platform for sharing and contributing multimedia associated with live events.

Live streaming is used herein to refer to delivery and/or receipt of content in real-time, as events happen, or substantially in real time, as opposed to recording content to a file before being able to upload the file to a media server, or downloading the entire file to a device before being able to watch and/or listen to the content. Streaming media is used herein to refer to multimedia (e.g., digital video and/or audio media) that is delivered between two or more network-connected devices in real time or substantially in real time. Streaming may apply to continuously updated media content other than video and audio including, but not limited to, a live ticker, closed captioning, and real-time text. An end-user (e.g., a viewer) may watch and/or listen to media streamed over a network (e.g., the Internet) using a user output interface such as a display and/or over a speaker communicatively coupled with, for example, a desktop computer, notebook or laptop computer, smart television, set-top box, Blu-Ray™ player, game console, digital media player, smartphone (e.g., iOS or Android), or another network-connected interactive device.

In some implementations, a network platform receives and provides multimedia (e.g., digital video content and/or digital audio content) associated with a live event. The multimedia may be captured by one or more broadcasters present at the live event. A broadcaster present at the live event may stream video and/or audio content to the network platform in real time or substantially in real time during the live event. For example, a broadcaster may capture video of a sporting event, such as a local high school football game, using a video camera, smartphone camera, etc. The video may include audio and/or visual commentary from the broadcaster. One or more viewers (either present or not present at the event) may stream video and/or audio of the event to watch and/or listen in real time or substantially in real time during the live event to the broadcaster's commentary. Alternatively, a broadcaster present at the live event may record video and/or audio content for delayed streaming or uploading to the network platform during or after the live event, and a viewer may download the broadcaster's recording of the live event and the video and/or audio commentary for delayed viewing and/or listening.

In some implementations, a broadcaster may or may not be present at a live event to still generate multimedia content (broadcaster commentary) associated with the event during the event. For example, a broadcaster may generate audio or visual content about the event while simultaneously following the event via a live broadcast by a third party (e.g., television, radio, Internet, etc.). The multimedia content may or may not include or be integrated with video and/or audio from the event itself.

In some implementations, a network platform is capable of integrating user-generated (broadcaster-generated) multimedia with real-time data (e.g., "event information") collected by the user or a third party. For example, a live competitive event may be integrated with scores for the event. Other real-time data may include but is not limited to alerts, statistics, trivia, polls, news, broadcaster and/or viewer messages, and/or advertising associated with or relevant to the event, a participant in the event, a location of the event, a date/time of the event, etc. In one implementation, a network platform allows a user to select content, for example, news articles, and create onscreen elements for simultaneous viewing of the content.

Audio and/or visual indications and content may be integrated with user-generated multimedia for simultaneous presentation. The presentation may be in real-time or substantially in real-time. For example, audio indications may be presented with digital video media, and/or visual content may be presented with digital audio media. In some implementations, audio and/or visual indications and content are presented simultaneously with digital audio and/or video media using multiple tracks and/or display frames or overlays. For example, digital video media of a basketball game or of a broadcaster providing play-by-play audio commentary for the game may be displayed with an overlay of a real-time scoreboard and/or ticker. Alternatively, the real-time scoreboard and/or ticker may be presented in a separate frame.

Audio and/or visual indications and content may be modifiable and/or interactive. For example, traditional news and sports broadcasting may insert audio and/or visual indications and content into an outgoing digital audio and/or video media stream. The receiving client devices have been assumed to be "dumb," that is, only capable of displaying the audio and/or video media as received. In contrast, in inventive implementations disclosed herein "smart" client devices allow audio and/or visual indications and content to be rendered on the client side, which allows for real-time modification and interaction with viewers and/or listeners.

That is, client-side rendering allows for interactivity with elements and enhanced features not available to traditional broadcasting.

FIG. **1**A is a block diagram of a system according to one inventive implementation, including multiple client devices (e.g., broadcaster client devices **100**A and **100**B, viewer client devices **200**A, **200**B, **200**C and **200**D), broadcast/viewing servers and memory storage devices **1000** (e.g., serving as the network platform noted above), an event information provider **55**, one or more news feeds (RSS feeds) **65**, and a digital distribution platform (app store) **75** all communicatively coupled via the Internet **50**. Each of the client devices **100**A, **100**B, **200**A, **200**B, **200**C, **200**D may download from the digital distribution platform **75** an app or software program that becomes resident on the client device (i.e., a client app) and performs at least some of the various broadcaster and viewer functionality described herein in connection with broadcasting live streams of digital content and viewing copies of broadcasted live streams, exchanging chat messages amongst broadcasters and one or more viewers, logging system events and providing system event messages to broadcasters and viewers, collecting and maintaining/updating event information and providing event information to broadcasters and viewers in a synchronized manner, providing and updating various animation and special effects graphics, and replaying of recorded streams.

Although FIG. **1**A illustrates two broadcaster client devices and four viewer client devices, it should be appreciated that various numbers of client devices (broadcaster client devices and viewer client devices) are contemplated by the systems, apparatus and methods disclosed herein, and those shown in FIG. **1**A are for purposes of illustration. More specifically, a given broadcaster may have virtually any number of viewers using respective viewer client devices to receive copies of the broadcaster's live stream of digital content via the servers and memory storage devices **1000**; similarly, the system may accommodate virtually any number of broadcasters providing live streams of digital content to the servers and memory storage devices **1000**, wherein each broadcaster has multiple viewers receiving copies of the broadcaster's live stream of digital content. In the example shown in FIG. **1**A, a first broadcaster client device **100**A provides a first live stream of digital content **102**A, and a first plurality of viewer client devices **200**A and **200**B (grouped by a first bracket) receive respective copies **202**A and **202**B of the first broadcaster's live stream of digital content. Similarly, a second broadcaster client device **100**B provides a second live stream of digital content **102**B, and a second plurality of viewer client devices **200**C and **200**D (grouped by a second bracket) receive respective copies **202**C and **202**D of the second broadcaster's live stream of digital content. With respect to events or news that may be germane to a given broadcaster's live stream of digital content, the broadcast/viewing servers and memory storage devices **1000** may retrieve various event information from the event information provider **55** (e.g., STATS LLC), and various news from news feeds (RSS) **65**, and in turn convey various event information and/or news to one or more client devices.

As discussed in further detail below, a variety of digital content format and transmission protocols are contemplated herein for the broadcaster live streams **102**A and **102**B output by the broadcaster client devices **100**A and **100**B respectively, as well as the copies of the live streams **202**A, **202**B, **202**C and **202**D received by respective viewer client devices **200**A, **200**B, **200**C and **200**D. For example, the first broadcaster client device **100**A may be a mobile broadcaster client device (e.g., a smartphone) and output a live stream of digital content **102**A having an H.264 MPEG-4 Advanced Video Coding (AVC) video compression standard format, via real time messaging protocol (RTMP) transport for continuous streaming over the Internet (e.g., via a persistent connection to a first media server of the servers and memory storage devices **1000**). The second broadcaster client device **100**B may be a web-based device (e.g., a desktop computer) and output a live stream of digital content **102**B having a VP8 video compression format, transmitted via the web real-time communication (WebRTC) protocol for continuous streaming over the Internet (e.g., via a persistent connection to a second media server of the servers and memory storage devices **1000**). The copies of the live streams **202**A, **202**B, **202**C and **202**D may be transmitted by the servers and memory storage devices **1000** as continuous streams using RTMP or WebRTC, or using segmented and/or adaptive bitrate (ABR) protocols (e.g., Apple's HTTP Live Streaming "HLS;" Microsoft's HTTP Smooth Streaming "MSS;" Adobe's HTTP Dynamic Streaming "HDS;" standards-based ABR protocol "MPEG-DASH").

FIG. **1**B illustrates a display **250** of an example viewer client device **200**A in the system of FIG. **1**A, showing various displayed content according to some inventive implementations. It should be appreciated that one or more elements of the various content discussed in connection with FIG. **1**B similarly may be provided on the display of a broadcaster client device. In the example of FIG. **1**B, a broadcaster is providing video-based commentary relating to a live sporting event, and the display **250** of the viewer client device **200**A includes various content elements including the broadcaster's video-based commentary **252**, event information **254** relating to the live sporting event about which the broadcaster is providing the video-based commentary, chat messages **258** from one or more viewers consuming the broadcaster's video-based commentary, and various graphics, special effects and/or animation elements **256** (e.g., some of which are rendered in a "lower third" of the display **250**).

More specifically, as shown in FIG. **1**B, the client device **200**A renders in the display **250** (pursuant to execution of a client app or software program) a first broadcaster's video-based commentary **252**. As discussed above in connection with FIG. **1**A, the first broadcaster's video-based commentary **252** is codified in a live stream of digital content **102**A provided by the first broadcaster client device **100**A to the servers and memory storage devices **1000**, and a copy **202**A of the first broadcaster's live stream is received by the viewer client device **200**A from the servers and memory storage devices **1000**. The display also includes event information **254** in the form of a "scorebug," wherein the scorebug includes indicators for the teams participating in the live sporting event, score information for the live sporting event, and event status (e.g., time clock, period or quarter, etc.). In various implementations discussed in further detail below, the scorebug may be animated, may include one or more special effects graphics elements, and/or may be interactive (e.g., the viewer may press or thumb-over one or more portions of the scorebug to launch further graphics or animations, receive additional information about the live sporting event, or navigate to another Internet location to receive additional information relating to the live sporting event).

The display **250** in FIG. **1**B also includes lower-third content **256** comprising additional graphics, special effects and/or animation elements which similarly may be interactive; such elements may include a broadcaster-selected title

for the broadcast, as well as text commentary from the broadcaster or event-related news. Additionally, as shown in the left portion of the display **250**, the display may include one or more chat messages **258** from different viewers of the broadcaster's video-based commentary, including responses from the broadcaster themselves; as seen in FIG. **1**B, the chat messages **258** may include the name of the viewer, a viewer photo, and the chat message content itself.

In some implementations, the network platform provided by the servers and memory storage devices **1000** maintains user profiles for broadcasters and viewers. Each user profile may be associated with, for example, a user email address, user device, or other unique identifier. Each user profile interface (e.g., "page" such as a webpage) may include and/or be customized with content (e.g., a profile photo, descriptive text, user-generated multimedia, favorite team imagery, etc.). In some implementations, the network platform further allows for the creation of "team" profiles; for example, event participants (e.g., individuals, groups, parties, teams, bands, schools, etc.) may share a "team" profile, wherein the team profile interface (e.g., "page" such as a webpage) may aggregate relevant content (e.g., news or current events about a particular event or team, such as polls, trivia, photo galleries, etc.) and provide further opportunities for users to contribute and connect with each other. The network platform may provide user preference options to further define a team profile interface with recommendations and/or alerts specific to a particular user (e.g., to prominently feature recent activity of a particular user).

With respect to social media-related features, as noted above the network platform provides chat capabilities such that users may engage in live public and/or private chat sessions. For example, in some implementations, users may request permission (or be allowed) to send each other private and/or public messages (e.g., direct messages). Furthermore, users may be able to purchase private and/or public virtual gifts (e.g., digital images of beers, penalty flags, etc., or profile/content enhancements like ticker tape) or provide "sponsorships" for other users. Public gifts received by a user may be displayed on the user's profile and/or with his or her content.

In some implementations, users are able to publicly and/or privately comment on, rate, "like," or otherwise indicate their opinions on live events, event-associated topics, user profiles, team profiles, and user-generated content. Users may be able to use #hashtags within their messages, chat sessions, comments, and/or other activity to link to messages, chat sessions, comments, and/or other activity happening among other users and/or teams. Users may be able to use @ symbols within their messages, chat sessions, comments, and/or other activity to tag other users, event participants, and teams.

In some implementations, a network platform provides a directory of live events. The directory interface may be presented as a listing, drop-down menu, keyword search bar, etc. The directory interface may include and/or distinguish between different categories of events. For example, the directory interface may include and/or distinguish between events that are scheduled, underway, and/or completed. The directory interface also may include and/or distinguish between different or particular types of events (e.g., live sports versus live music, baseball versus hockey, professional versus collegiate, National League versus American League, etc.); different or particular participants in the events (e.g., team, coach, athlete, owner, school, etc.); and/or different or particular locations of the events (e.g., country, region, state, county, town, district, etc.). As discussed in greater detail below, in one implementation a dedicated control server of the network platform periodically retrieves a variety of event information from one or more event information providers (e.g., for sports events, ESPN, STATS LLC), and populates a database of the network platform with information on available events so as to provide the directory of live events to a user.

In some implementations, the network platform may provide user preference options to further define an event directory interface with recommendations and/or alerts specific to a particular user. The network platform may request the location of a user or permission to access the geolocation of the user's device in order to recommend events nearby. The network platform may track and interpret patterns in the user's use of the platform to predict and recommend events specific to the user.

In some implementations, after a user selects an event, the network platform provides a directory of other users who are present at the event and/or generating media associated with the event. The directory interface may be presented as a listing, drop-down menu, keyword search bar, etc. Selection of another user from the event-specific directory allows connection to, communication with, and/or access to media generated by that user. Thus, a user is able to discover and connect with similar users. The network platform may provide user preference options to further define a user directory interface with recommendations and/or alerts specific to a particular user. For example, in some implementations, users can discover other users based in part on one or more of the location of respective users, an event about which the broadcaster is providing commentary, a title of a broadcaster's live stream, and topics or other users that have been identified (e.g., in chat messages relating to a given broadcaster's live stream and/or a particular user's profile, using #hashtags or @ symbols).

In some implementations, the popularity of an event and/or broadcaster is monitored, displayed, and/or used in real-time or substantially in real-time. For example, a number of video servers may be scaled based on demand and/or usage by client devices, including broadcasters and/or viewers. Worker servers may be used for distributed monitoring and capturing screenshots/thumbnails of video streams. In another example, client media source selection of live stream copies, such as Real-Time Messaging Protocol (RTMP) versus HTTP Live Streaming (HLS), may be based on demand and/or usage levels (e.g., number of viewers requesting copies of a given broadcaster's live stream, capacity of media servers and/or content delivery network).

II. Servers and Memory Storage Devices

Having provided an overview of the information flow and general functionality enabled by the various elements shown in FIG. **1**A, additional details of the servers and memory storage devices **1000** are now discussed, with reference initially to FIG. **2**.

In particular, FIG. **2** is a block diagram providing another perspective of the system shown in FIG. **1**A, showing example communication connections between the broadcaster client devices **100**A and **100**B and the servers and memory storage devices **1000**, example connections between the servers and memory storage devices **1000** and the viewer client devices **200**A and **200**C, and some structural details of the servers and memory storage devices **1000**. Some of the broadcaster/viewer client devices that are mobile devices (e.g., smartphones) have downloaded a client app **5000** (e.g., from the digital distribution platform or

app store **75** shown in FIG. **1**A) which is resident in memory of the client device and executed by a processor of the client device. For purposes of simplifying the illustration, only the viewer client devices **200**A and **200**C explicitly show the client app **5000** resident on the client devices; it should be appreciated, however, that one or more mobile broadcaster client devices also have the client app **5000** installed thereon.

As shown in FIG. **2**, in one inventive implementation the servers/memory storage devices **1000** include one or more web servers **700** (also referred to herein as a "web server pool") that support an Applications Programming Interface (API) to facilitate communications between the servers/memory storage devices **1000** and one or more mobile broadcaster/viewer client device executing the client app **5000**, and also facilitate communications to and from web-based client devices (that access the web server(s) via a web portal at a particular URL). In this role, as discussed in further detail below, much of the instructive communication between the client devices and the servers/memory storage devices **1000** occurs via the web server(s) **700**. For example, it is via the web server(s) **700** that client devices create new live streams for broadcast and get access to media servers, receive access to view other broadcasters' live streams via one of multiple different media sources, receive event information associated with broadcasters' live streams and send and receive chat messages, log on and create or update user profiles (or other profiles such as team profiles), and access other social media-related functionality (e.g., digital gift giving) to interact with other users. The web server(s) **700** are communicatively coupled to a memory system **400** that includes a database **420**, data storage **440**, and one or more memory caches **460** to store various information (e.g., user profile information, stream information, event information, recorded live streams, etc.) germane to the operation of the servers and memory storage devices **1000** and the various client devices.

The servers/memory storage devices **1000** further comprise a plurality of media sources **300** (e.g., computer servers including one or more processors, memory, and one or more communication interfaces) that receive a live stream of video-based commentary from a given broadcaster client device, and provide copies of the live stream of video-based commentary to one or more viewer client devices. As shown in FIG. **2**, in one implementation the media sources **300** are communicatively coupled to the memory system **400**, and may comprise one or more Real Time Messaging Protocol (RTMP) media servers **320**, an RTMP Content Delivery Network (CDN) **340** (which itself includes a plurality of content delivery network servers), one or more WebRTC media servers **360**, and an inventive HTTP Live Streaming (HLS) caching and amplifying server architecture **380**. Additional details of the media sources **300** are discussed below in connection with FIGS. **3** through **6**, and particular details of media server processes (performed by the RTMP media servers **320** and the WebRTC media servers **360**) are discussed below in connection with FIGS. **5**A, **5**B and **5**C. As also discussed below, in one implementation the web server(s) **700** select a particular media server of the media sources **300** to which a given broadcaster connects to provide the broadcaster's live stream of digital content, and the web server(s) **700** also select a particular media source of the media sources **300** to which a given viewer connects to receive a copy of a given broadcaster's live stream; further details of a broadcast media server selection algorithm and a viewer stream source selection algorithm implemented by the web server(s) **700** are provided below in connection with FIGS. **6** and **7**.

The servers/memory storage devices **1000** shown in FIG. **2** further comprise a control server **500** coupled to the memory system **400**, the event information provider **55**, and the news feeds (RSS) **65** (e.g., via the Internet). In one aspect, the control server **500** periodically retrieves various event information from the event information provider **55** and/or news from the news feeds **65** that is germane to respective broadcasters' video-based commentary. In another aspect, the control system **500** may store at least some portion of retrieved event information and/or news in the memory system **400**. More generally, as discussed below in connection with FIG. **10**, the control server **500** implements a number of services/processes that govern functionality of other servers and devices in the servers/memory storage devices **1000**; examples of such control system services/processes include, but are not limited to: an RTMP media server scaling process to add or remove servers from the one or more RTMP media servers **320** of the media sources **300** (see FIG. **11**); an RTMP CDN server scaling process to add or remove servers from the RTMP CDN **340** of the media sources **300** (see FIG. **12**); a live stream and media server watchdog process (see FIGS. **13**-**14**); an event data ingress process (see FIG. **15**); a live event data monitor process (see FIG. **16**); an asynchronous task processor (see FIG. **17**); and a live stream thumbnail/screenshot acquisition process (see FIG. **18**).

With reference again to FIG. **2**, the servers/memory storage devices **1000** further comprise one or more socket servers **600** communicatively coupled to the web server(s) **700** and the control server **500**. In one aspect, the socket server(s) **600** facilitate communication, to one or more broadcaster client devices and one or more viewer client devices, of synchronized event information retrieved by the control server **500** and associated with video-based commentary relating to a particular event. In particular, one or more sockets of the socket server(s) dedicated to the particular event allow respective client devices to establish an event information channel with the socket server(s), such that the event information (e.g., in the form of "event messages") is shared in a synchronized manner by all broadcasters/viewers following the particular event.

In FIG. **2**, the socket server(s) **600** also facilitate communication, between a given broadcaster of a live stream of video-based commentary and corresponding viewers of copies of the live stream, of chat messages and/or system event information (also referred to collectively simply as "chat information") relating to the broadcaster's live stream. In particular, 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. Chat messages sent on a given chat/system event channel may be displayed as text on all broadcaster/viewer client devices connected to the socket(s) dedicated to the particular broadcaster's live stream, whereas system event information may be received (but not necessarily displayed itself) by all client devices connected to the socket(s) dedicated to the particular broadcaster's live stream, and provide the client device with relevant data or instructions to take some action. As discussed further below, examples of the types of system event information or "system messages" that may be broadcast by the socket(s)

dedicated to the particular broadcaster's live stream include, but are not limited to, indications of viewers joining or leaving a broadcast, an indication of a new follower of a broadcaster, indications relating to the purchase of digital gifts and types of digital gifts (which may cause some display or audio event on the client device), indications relating to "likes" (e.g., cheers, handclaps, or applause icons, or audio of crowds cheering), and other data/instructions relating to various social networking functionality.

In one aspect, connections between a given client device and a particular socket of a socket server are persistent authenticated connections (e.g., with IP-based fingerprint identifiers for anonymous users). The authenticated connection allows the servers and media storage devices **1000** to track how many users are connected to a particular socket at any given time (and hence how many users are viewing a copy of a particular broadcaster's live stream, and/or how many users are viewing a copy of a live stream relating to a particular event). In another aspect, the various "messages" (e.g., event messages, chat messages, system messages) that are carried on the respective channels between a given client device and corresponding sockets of the socket server(s) are data packets including various event information, chat to be displayed, or system events (e.g., "new viewer," "disconnected viewer," "stream muted, "stream ended").

With reference again for the moment to FIG. **1**A, recall that in the example arrangement depicted in FIG. **1**A a first broadcaster client device **100**A provides a first live stream of digital content **102**A, and a first plurality of viewer client devices **200**A and **200**B (grouped by a first bracket) receive respective copies **202**A and **202**B of the first broadcaster's live stream of digital content. Similarly, a second broadcaster client device **100**B provides a second live stream of digital content **102**B, and a second plurality of viewer client devices **200**C and **200**D (grouped by a second bracket) receive respective copies **202**C and **202**D of the second broadcaster's live stream of digital content. Turning now again to FIG. **2**, and taking only the viewer client devices **200**A and **200**C into consideration for purposes of illustration, the example implementation shown in FIG. **2** contemplates that the first broadcaster is providing video-based commentary about a first live sporting event, and the second broadcaster is providing video-based commentary about a second (different) live sporting event, such that the first viewer client device **200**A receives the copy **202**A of the first broadcaster's live stream of digital content **102**A relating to the first sporting event (and provided by the first broadcaster client device **100**A), and that the second viewer client device **200**C receives the copy **202**C of the second broadcaster's live stream of digital content **102**B relating to the second sporting event (and provided by the second broadcaster client device **100**B). Also, in the example of FIG. **2**, the first broadcaster's live stream **102**A is an RTMP stream received by the RTMP media server(s) **320**, and the second broadcaster's live stream **102**B is a WebRTC stream received by the WebRTC media server(s) **360**. The media sources **300** provide the copy **202**A of the first broadcaster's live stream **102**A to the first viewer client device **200**A via a first video Internet communication channel **204**A, and provide the copy **202**C of the second broadcaster's live stream **102**B to the second viewer client device **200**C via a second video Internet communication channel **204**C (further details of the role of the web server(s) **700** in selecting a particular media source of the media sources **300** to which each viewer client device connects to establish a video Internet communication channel are discussed below in connection with FIGS. **6** and **7**).

In the example of FIG. **2**, as noted above the control server **500** periodically retrieves, via the Internet and from the event information provider **55**, first event information **502**A germane to the first live sporting event, wherein the first event information includes at least first score information **504**A for the first live sporting event. The control server further retrieves second event information **502**B germane to the second live sporting event, wherein the second event information includes at least second score information **504**B for the second live sporting event. The control server passes at least the first score information **504**A and the second score information **504**B to the socket server(s) **600**. In turn, the socket server(s) **600** establish one or more first event sockets **602**A dedicated to the first event information and one or more second event sockets **602**B dedicated to the second event information.

As discussed further below, the web server(s) **700** provide to the first viewer client device **200**A a first event identifier (a first EventID) that corresponds to the first event socket **602**A; the web server(s) **700** also provide to the second viewer client device **200**C a second event identifier (a second EventID) that corresponds to the second event socket **602**B. The first viewer client device **200**A uses the first EventID to connect to the first event socket **602**A (e.g., via a first URL including the first EventID in a path of the URL), and the second viewer client device **200**C uses the second EventID to connect to the second event socket **602**B (e.g., via a second URL including the second EventID in a path of the URL). The first score information **504**A is then transmitted to the first viewer client device **200**A via a first event information Internet communication channel **206**A between the first event socket **602**A and the first viewer client device **200**A, and the second score information **504**B is transmitted to the second viewer client device **200**C via a second event information Internet communication channel **206**C between the second event socket **602**B and the second viewer client device **200**C.

In a manner similar to that described above in connection with the first and second event information, in the example of FIG. **2** chat messages and other system event information ("chat information") may be distributed to viewers of each broadcaster via respective dedicated sockets of the socket server(s) **600**. In particular, the socket server(s) **600** similarly establish one or more first chat/system event sockets **604**A dedicated to the first broadcaster's live stream of digital content **102**A and one or more second chat/system event sockets **642**B dedicated to the second broadcaster's live stream of digital content **102**B. The web server(s) **700** provide to the first viewer client device **200**A a first stream identifier (a first StreamID) that corresponds to the first chat/system event socket **604**A; the web server(s) **700** also provide to the second viewer client device **200**C a second stream identifier (a second StreamID) that corresponds to the second chat/system event socket **604**B. The first viewer client device **200**A uses the first StreamID to connect to the first chat/system event socket **604**A (e.g., via a first URL including the first StreamID in a path of the URL), and the second viewer client device **200**C uses the second StreamID to connect to the second chat/system event socket **604**B (e.g., via a second URL including the second StreamID in a path of the URL). The first chat information **210**A is then transmitted to the first viewer client device **200**A via a first chat/system event Internet communication channel **208**A between the first chat/system event socket **604**A and the first

viewer client device **200**A, and the second chat information **210**B is transmitted to the second viewer client device **200**C via a second chat/system event Internet communication channel **208**C between the second chat/system event socket **604**B and the second viewer client device **200**C.

For purposes of simplifying the illustration in FIG. **2**, the broadcaster client devices **100**A and **100**B are shown only providing respective live streams **102**A and **102**B directly to different media servers **320** and **360**; however, it should be appreciated that the broadcaster client devices **100**A and **100**B have additional communication connections to the socket server(s) **600** and the web server(s) **700**, similar to those shown in FIG. **2** between the example viewer client devices **200**A and **200**C and the socket server(s) **600** and web server(s) **700**, so that the broadcaster client devices may similarly receive event information and chat information on different communication channels respectively dedicated to the event information and chat information.

In view of the foregoing, it may be appreciated from FIG. **2** that, in one example implementation, there are three different communication channels between a given broadcaster/viewer client device and the broadcast/viewing servers and media storage devices **1000**, namely: 1) a video communication channel (e.g., **204**A, **204**C) between the client device and the media sources **300** to receive a copy of a broadcaster's live stream of digital content; 2) an event information communication channel (e.g., **206**A, **206**C) between the client device and one or more particular sockets of the socket server(s) **600** dedicated to a particular event; and 3) a chat/system event communication channel (e.g., **208**A, **208**C) between the client device and one or more particular sockets of the socket server(s) **600** dedicated to a particular broadcaster's live stream of digital content.

In the example of FIG. **2**, the first and second broadcasters provide to their respective viewing audiences video-based commentary regarding different live sporting events. However, as discussed elsewhere in this disclosure, it should be appreciated that the events about which the broadcasters provide video-based commentary are not limited to live sporting events, but may relate to a wide variety of other events, news, and/or particular topics of interest. Additionally, it should be appreciated that the first and second broadcasters (and additional broadcasters) may provide to their respective viewing audiences video-based commentary about the same live event; in this case, the servers and media storage devices **1000** provide the appropriate connectivity such that viewers of the same live event may effectively switch between different broadcasters' video-based commentary about the event, participate in different chat information exchanges associated with each broadcaster's live stream, and all share the same event information in a synchronized manner.

In particular, with reference again to the example of FIG. **2**, consider an implementation in which both the first broadcaster's live stream of digital content **102**A and the second broadcaster's live stream of digital content **102**B include the broadcasters' respective video-based commentary about the first live sporting event. In this situation, the web server(s) **700** would provide to both the first viewer client device **200**A and the second viewer client device **200**C the first event identifier (the first EventID) that corresponds to the one or more first event sockets **602**A of the socket server(s) **600**, and both of the first viewer client device **200**A and the second viewer client device **200**B would use the first EventID to connect to the one or more first event sockets **602**A (e.g., via a first URL including the first EventID in a path of the URL). In this manner, the first score information **504**A would then be transmitted to both the first viewer client device **200**A via the first event information Internet communication channel **206**A between the one or more first event sockets **602**A and the first viewer client device **200**A, and to the second viewer client device **200**C via a second event information Internet communication channel **206**C between the one or more first event sockets **602**A and the second viewer client device **200**C. Thus, both of the viewer client devices in this scenario would receive the same event/score information for the first live sporting event in a synchronized manner from the socket server(s).

At the same time, however, the respective viewer client devices **200**A and **200**C would be connected to different chat/system event sockets of the socket server(s) corresponding to the different broadcasters' live streams; in particular, the web server(s) **700** would provide to the first viewer client device **200**A the first stream identifier (the first StreamID) that corresponds to the first chat/system event socket **604**A and provide to the second viewer client device **200**C the second stream identifier (the second StreamID) that corresponds to the second chat/system event socket **604**B. As discussed in the previous example, the first viewer client device **200**A would use the first StreamID to connect to the first chat/system event socket **604**A (e.g., via a first URL including the first StreamID in a path of the URL), and the second viewer client device **200**C would use the second StreamID to connect to the second chat/system event socket **604**B (e.g., via a second URL including the second StreamID in a path of the URL). The first chat information **210**A would then be transmitted to the first viewer client device **200**A via a first chat/system event Internet communication channel **208**A between the first chat/system event socket **604**A and the first viewer client device **200**A, and the second chat information **210**B would be transmitted to the second viewer client device **200**C via a second chat/system event Internet communication channel **208**C between the second chat/system event socket **604**B and the second viewer client device **200**C.

FIG. **3** is a block diagram showing additional details of various interconnections between the respective components of the servers and memory storage devices **1000** shown in FIG. **2**, according to some inventive implementations. In the example of FIG. **3**, some of the components of the servers and memory storage devices (e.g., **1000**A) are hosted by a first web hosting service (e.g., Amazon Web Services AWS), while one or more other components of the servers and memory storage devices (**1000**B) may be hosted by a different web hosting service and/or generally accessible via the Internet. In yet other implementations, a single web hosting service may host all of the servers and memory storage devices. In addition to the various components shown in the example of FIG. **2**, FIG. **3** also shows that the servers and memory storage devices **1000** may further include a transcoder server pool **800** (e.g., that may be employed for transcoding of recordings of a given broadcaster's live stream of digital content, for later replay via adaptive bitrate protocols), an asynchronous queue **850** (e.g., for queuing of various messages and instructions to be acted upon by an asynchronous task processor implemented by the control server **500**), and a gateway NAS server **870** (e.g., to facilitate communications between a WebRTC media server pool and other elements of the servers and memory storage devices **1000**A that may be hosted by the first web hosting service). Additionally, FIG. **3** illustrates that the database **420** may include a main database and multiple database shards, in which portions of data are placed in relatively smaller shards, and the main database acts as a directory for

the database shards (in some implementations, the main database also stores some de-normalized data, for example, to facilitate cross-server searching).

III. Technological Solutions to Improve Computer Network Functionality, Increase Computer Processing Efficiency and Reduce Computer Memory Requirements

In developing the inventive systems, apparatus and methods disclosed herein, including the servers and memory storage devices **1000** shown in FIGS. **2** and **3** as well as the client app **5000** executed by mobile client devices, the Inventors recognized and appreciated multiple technological problems with conventional techniques for transmission of digital content via the Internet. As introduced above and discussed in further detail below, the Inventors have addressed and overcome these technological problems with innovative technological solutions to effectively realize the various technical features described herein. Examples of these technological solutions include, but are not limited to, improving computer network functionality (e.g., improving the speed of content transfer from broadcaster devices to viewer devices and synchronization of various content amongst multiple client devices), and improving processing efficiency of broadcaster and viewer client devices via execution of the client app **5000**, while at the same time reducing memory storage requirements for the client app **5000** on the client devices.

More specifically, examples of the technological problems addressed by the inventive solutions provided by the servers and memory storage devices **1000** and client app **5000** include, but are not limited to: 1) how to provide relatively low latency copies of live streams of broadcaster digital content to multiple viewers of each of multiple broadcasters (e.g., broadcaster-to-viewer delay time on the order of ten seconds or less, or on the order of two-to-three seconds or less), and with relatively high quality and reliability (e.g., high definition HD and high bit rate, such as 2 to 5 megabits per second); 2) how to synchronize such low latency and high quality copies of broadcaster live streams of digital content with event information associated with the digital content (as well as chat information associated with a given broadcaster) amongst the multiple viewers of each broadcaster, irrespective of the number of viewers (e.g., 10 viewers, 1,000 viewers, or 10,000 viewers); 3) how to allow different classes/types of viewers (e.g., VIP users, premium subscribers, media professionals, registered users, anonymous users, web/desktop users, mobile users), and increasing numbers of viewers, to flexibly access each broadcaster's content with different live streaming formats (e.g., continuous streaming protocols such as real time messaging protocol or "RTMP," web real-time communication or "WebRTC;" segmented protocols such as HTTP live streaming or "HLS," HTTP Smooth Streaming or "MSS," HTTP Dynamic Streaming or "HDS," standards-based ABR protocol "MPEG-DASH") and with different qualities of service; 4) how to effectively render "studio-quality" screen animations and special effects graphics (e.g., including "scorebugs" for sporting events) on displays of mobile client devices via a client app with a small memory footprint (e.g., less than 100 megabytes, such that the client app is downloadable via cellular networks); and 5) how to provide for viewing of a recording of a broadcaster's live stream as if the viewer was watching the live stream in essentially real-time (e.g., while recreating chat messages and event information updates). Various aspects of the technological solutions to these respective technological problems are discussed in turn below.

1) Latency Considerations

With respect to latency considerations, the inventive systems, methods and apparatus disclosed herein contemplate particular parameters for the generation of a live stream of digital content by a broadcaster client device so as to induce only relatively low "client side" latency. To this end, in example implementations the client app **5000** installed and executing on a given client device selects an appropriate keyframe interval (e.g., 30 frames) for generating a broadcaster's live stream of digital content to ensure relatively low client side-induced end-to-end digital content latency.

In other aspects relating to reducing latency, particular parameters and techniques for handling live streams are contemplated for the servers and memory storage devices **1000** disclosed herein (e.g., adjusting buffer sizes and transcoder settings in media servers; employing hardware-accelerated transcoding of broadcaster live streams via graphic card processing to provide for adaptive bitrate copies of live streams). Furthermore, in some example implementations, the RTMP CDN **340** shown in FIGS. **2** and **3** comprises an innovative auto-scaling RTMP CDN server pool, coupled to a media server pool that receives live streams from respective broadcasters (e.g., either RTMP or WebRTC), to facilitate delivery of low-latency live streams to a larger number of multiple viewers. Additionally, for RTMP broadcasters, the RTMP media server(s) **320** in some implementations is/are on the same network as the RTMP CDN **340** (e.g., the RTMP media server(s) are communicatively coupled to the RMTP CDN servers as a virtual private network (VPN), see VPN **330** in FIG. **6**) so as to facilitate low latency communications. For WebRTC broadcasters, although in some implementations the WebRTC media server(s) **360** may not be hosted by the same service as the RTMP CDN **340** (e.g., see FIG. **3**), the WebRTC media server(s) are coupled to the RTMP CDN via high speed/low latency connections. The RTMP CDN servers essentially make further copies of transcoded live streams received from the media server (e.g., without any other processing or alteration) and pass on the respective further copies to multiple viewers ("direct pass-through amplification"). In this manner, the RTMP CDN servers introduce appreciably low latency (e.g., on the order of less than 150 milliseconds) and facilitate a significantly greater number of viewers than could be otherwise served by the media server itself. These exemplary aspects (as well as other aspects discussed in further detail below) provide for appreciably low latency introduced by the media servers and RTNIP CDN (e.g., on the order of about 500 milliseconds or even less) and client-introduced digital content latency (e.g., on the order of about one-to-two seconds for continuous streaming consumers).

2) Synchronization of Live Streams and Event Information

Yet another technical implementation challenge overcome by the inventive concepts disclosed herein relates to the display of event information updates (if present, e.g., if the broadcast is associated with an event), as well as screen animations and other special effects graphics that may be generally associated with the video and/or audio associated with a live stream, in a manner that is synchronized across multiple live streams with appreciably low latency. This is a particularly relevant consideration given that the systems, apparatus and methods disclosed herein are contemplated in

some implementations as supporting multiple broadcasters providing video-based commentary for the same event, and each of these broadcasters may have multiple viewers of their broadcast—and thus, the technical challenge is to provide the same event information, and periodic updates to this event information, in a synchronized and low-latency manner to all of these broadcasters and viewers interested in following the same event. In exemplary implementations (e.g., as discussed above in connection with FIG. **2**), this technical challenge is overcome with technological solutions implemented on both the client devices and the server architecture to which the client devices are communicatively coupled involving the use of multiple communication channels respectively dedicated to video/audio content from a given broadcaster, event information germane to an event about which any broadcaster may be providing video-based commentary, and chat information (chat messages and/or system event messages) shared amongst the broadcaster and their associated viewers.

In various inventive implementations disclosed herein (e.g., as introduced above in connection with FIG. **2**), event information and updates to event information are provided to broadcaster client devices and viewer client devices via a socket-based "event information channel" dedicated to the event, and separate from the copy of the live stream of video-based commentary provided on a "video channel." Thus, all viewers (and broadcasters) of the event, regardless of which live stream they may be generating or watching, connect to one or more sockets of a socket server that is/are dedicated to the event, such that all live streams relating to the event are similarly synchronized to event information and updates to same. Notably, 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.

The technical challenge of displaying event information and updates to same in a synchronized and low-latency manner amongst multiple viewers is also addressed in part by using a single control server **500** in the server and memory storage devices **500** to gather and parse live event information captured in real-time. For example, for sporting events, game information may be obtained by the single control server from a dedicated third-party provider (e.g., STATS LLC, which is a sports statistics, technology, data, and content company that provides content to multimedia platforms, television broadcasters, leagues and teams, fantasy providers, and players). This single point of entry of event information into the server architecture, as provided by the control server, prevents synchronization errors inherent in network communications. Once a change in event status has been detected (e.g., if a play clock updates), the control server provides these changes to the one or more sockets dedicated to the event (to which all viewers and broadcasters of video-based commentary regarding the event are communicatively coupled), resulting in a single synchronized update to all client devices and thereby significantly mitigating client-by-client latency and/or synchronization issues.

3) Flexible and Scalable Access to Broadcaster Content by Multiple Classes/Types of Viewers

The inventive systems, methods and apparatus disclosed herein and shown in FIGS. **2** and **3** further contemplate the ability to flexibly select the source of a copy of a broadcaster's live stream to be provided to respective multiple viewers from one of a number of possible media sources **300**, namely: 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 **360**. Thus, multiple live stream transmission formats, protocols, and access endpoints 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. As noted above, in some implementations the web server(s) **700** implement a 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. Further details of viewer stream source selection for respective viewer client devices are discussed further below in connection with FIGS. **6** and **7**.

Another salient element of the flexibility and scale-ability provided by the media sources **300** of the servers and memory storage devices **1000** shown in FIGS. **2** and **3** relates to the HLS caching and amplifying server architecture **360**. Conventionally, as would be readily appreciated by those of skill in the relevant arts, HLS is not designed to be cacheable at the server level, and hence synchronization issues arise in connection with providing multiple HLS copies of a live stream to respective viewers. In particular, in conventional implementations, each HLS copy of the live stream is somewhere in a "window" of time (an HLS "buffer length") relative to the original live stream (e.g., delayed from the original stream by some amount of time within an overall time window). This uncertainty results in the possibility of a first viewer of a first HLS copy of a live stream actually seeing the video content some time earlier than or later than a second viewer receiving a second HLS copy of the live stream, i.e., the respective viewers are not synchronized.

In exemplary implementations described herein, this technical problem is solved by employing an inventive HLS caching and amplifying server architecture **360**, which is discussed in further detail below in connection with FIGS. **8**, **9**A, **9**B, **9**C and **9**D. The HLS server architecture includes a "mother" server and one or more "child" servers, disposed between a media server and a content delivery network (CDN), in which the HLS mother server acts as a single "virtual viewer" from a given media server's perspective. Based on a single copy of an HLS file suite for a given broadcaster's live stream as provided by a media server and received by a mother caching server of the HLS server architecture, the mother server caches and passes on copies of the elements of the file suite (as requested) to one or more child servers, which in turn cache and pass on copies of the elements of the file suite to one or more geographically-distributed servers of a conventional (e.g., global) CDN (serving as an HLS CDN in tandem with the mother-child server architecture). In this manner, the mother and child servers of the HLS architecture act as caching and amplifying servers, so that identical HLS streams may be served from the HLS CDN server pool to multiple viewers of a given broadcast in a significantly narrower synchronization window than conventionally possible. In particular, in one example implementation discussed in greater detail below in

connection with FIGS. **6**A, **6**B, **6**C, and **6**D, all HLS viewers receiving a copy of a broadcaster's live stream via the HLS server architecture including a mother caching server and one or more child caching servers are at most less than one HLS file segment duration out of synchronization with each other; this phenomenon is referred to herein as "viewer segment concurrency." Based on the viewer segment concurrency provided by the inventive HLS server architecture, respective viewers of a given broadcast may be out of synchronization with one another by less than approximately one or two seconds at most.

4) Client-Side Rendering of On-Screen Interactive Animations, Special Effects and/or Event Information

By way of background, in conventional sports broadcasting, game information (also sometimes referred to as a "scorebug"), as well as screen animations and other special effects graphics, are hard-embedded into the live stream of the game broadcast itself that is received by viewers. Unlike conventional scorebugs, screen animations, and/or other special effects graphics that are hard-embedded into live streams of a sports broadcast, in various inventive implementations disclosed herein graphics and effects are generated by the client device itself, separate from a given broadcaster's video-based commentary, and then integrated with (e.g., superimposed or overlaid on) the broadcaster's video-based commentary when rendered on the display of the client device. As shown for example in FIG. **1**B, various graphics may be rendered on different portions of the display, for example, along a top or side of the display or in a "lower third" of the display.

For mobile client devices, the client app **5000** executing on the device is particularly configured to render a variety of "studio-quality" graphics while nonetheless maintaining a small file size for the client app (e.g., less than 100 megabytes, and in some instances from approximately 60-70 megabytes); this affords an exciting and dynamic broadcaster and viewer experience on mobile client devices, while still allowing the modestly-sized client app to be readily downloaded (e.g., from a digital distribution platform or "app store" **75**) to a client device via a cellular network. In some implementations, maintaining a modest file size for the client app while providing high-quality graphics, animations and other special effects is accomplished in part by designing animated graphics and special effects as a series of individual frames (still-frame images) that are hard-coded in the client app, and rendering the series of individual frames on the display in a "stop-motion" style according to an animation timer set in the client device (e.g., 15 frames per second). In some implementations, "sprite sheets" may be used for graphics elements; in yet other implementations, the transparency of individual frames may be set on a pixel-by-pixel basis as may be required in some applications to provide for suitable overlay on the broadcaster's video-based commentary.

In another aspect, client-side rendering of screen animations and/or other special effects graphics allows such animations and graphics to be user-interactive; for example, a user (broadcaster or viewer) on a client device may "select" a screen animation/special effect graphic (e.g., via a touch-sensitive display screen of the client device) and launch additional graphics or initiate some other functionality on the client device.

For example, as discussed above with respect to live events about which a given broadcaster may be providing video-based commentary, event information and updates to event information are provided to broadcaster client devices and viewer client devices via a socket-based "event information channel" dedicated to the event, and separate from the copy of the live stream of video-based commentary provided on a "video channel." Providing one or more sockets dedicated to the event information and separate from the live stream of video-based commentary provides for user-interactive features in connection with the event information, and/or the screen animations/special effects graphics incorporating the event information; for example, the user may select (e.g., thumb-over) the screen animation/special effect graphic including the event information and obtain access to additional (and in some cases more detailed) information relating to the event (e.g., a drill down on more granular event information, or a redirect to a web site or other app related to the particular event).

5) Replay of Recorded Broadcaster Live Streams with Recreated Chat Messages and Event Information Updates

Another technical implementation challenge addressed by the technological solutions disclosed herein relates to the ability of a viewer to watch a recording of a live stream generated by a broadcaster client device (also referred to herein as a "video replay" of the live stream, or simply "replay") as if the viewer was watching the live stream in essentially real-time (as it was being generated by the broadcaster client device), while also allowing the viewer to "seek" to different points in the video replay. In one aspect of video replay, the broadcaster themselves may assume the role of a post-broadcast viewer of the recorded broadcast.

In exemplary implementations, a technological solution for overcoming the technical implementation challenge of replaying a recorded live stream and also recreating various chat messages and event information updates (if present) as they occurred during the originally broadcast live stream is based, at least in part, on having the socket-based communication techniques act in a "fully-authenticated" fashion, for example, by dynamically creating "anonymous accounts" for non-registered or "anonymous" users. By creating such accounts for anonymous users, a replay log may be created that logs when any given viewer (as a registered user or anonymous user) joins and leaves a particular broadcast. Additionally, the replay log may include additional information, such as user-generated chat information, system messages, and event information updates, respectively synchronized with timestamps associated with the live stream as originally generated by the broadcaster client device.

During replay of a recording of the live stream, the viewer client device requests a segment of this replay log and, using the timestamps in the recording of the live stream, replays not only the digital content in the live stream but also recreates chat messages, system-related messages and event information updates (if present) in the same order and relative time of occurrence as if the viewer were watching the live stream in essentially real-time when originally broadcasted by the broadcaster. As the replay advances, the viewer client device requests additional segments of the log, keeping an in-memory buffer to smooth out any possible Internet connectivity issues. Such a replay log also allows for "seeking," i.e., when a viewer fast forwards or rewinds; under these seeking circumstances, the viewer client device may retrieve the appropriate segment(s) of the replay log for the new viewing point, and continue to not only replay the recording of the live stream from the new viewing point but also recreate (in the same order and relative time) chat messages, system-related messages and event information updates (if present) as if the viewer were watching the live stream in essentially real-time.

Having outlined some of the various technological solutions provided by the inventive systems, apparatus and methods disclosed herein to technological problems with conventional approaches to live streaming of digital content, the discussion now turns to additional details of respective components of the servers and memory storage devices **1000** shown in FIGS. **1**A, **2** and **3**, as well as the functionality of the client app **5000** executed by client devices.

IV. Broadcaster Media Server Selection

FIGS. **4**A and **4**B show a process flow diagram **450**A and **450**B illustrating a broadcast media server selection algorithm according to one inventive implementation, which in some examples may be performed by the web server(s) **700** shown in FIGS. **2** and **3**. As noted above, in one implementation a mobile broadcaster client device (e.g., a smartphone) outputs a live stream of digital content having an H.264 MPEG-4 Advanced Video Coding (AVC) video compression standard format, via real time messaging protocol (RTMP) transport for continuous streaming over the Internet, whereas a web-based broadcaster client device (e.g., a desktop computer) outputs a live stream of digital content **102**B having a VP8 video compression format, transmitted via the web real-time communication (WebRTC) protocol for continuous streaming over the Internet.

In the process shown in FIGS. **4**A and **4**B, the web server(s) **700** know whether the broadcaster client device requesting access to a media server is a mobile client (H.264/RTMP) or a web-based client (VP8/WebRTC) based on header information in the communications to the web server from the client device. For mobile clients, the web server provides access to (e.g., provides the address of an endpoint for) one of the RTMP media servers **320** of the media sources **300**, and for web-based clients generating VP8/WebRTC live streams of digital content, the web server provides access to one of the WebRTC media servers **360** of the media sources **300**. If a web-based client is connecting via Adobe Flash or other external software, the client may be treated similarly to the process for mobile clients.

In some implementations, multiple media servers of the RTMP media servers **320** are segregated into at least one VIP media server and at least one non-VIP media server; similarly, some of the WebRTC media servers **360** are segregated into at least one VIP media server and at least one non-VIP media server. A given broadcaster may be directed to a VIP or non-VIP media server based on their user status (e.g., as a VIP user), and/or the availability of a particular server (e.g., based on available server capacity, in terms of total utilized connection bandwidth to the media server). In one aspect, to allow for some headroom in media server capacity, the "ideal capacity" of the server may be taken as approximately 60% of the true maximum capacity of the media server. If all non-VIP media servers exceed ideal capacity (but are at less than true maximum capacity), the process may send an internal administrative message (e.g., via SMS or email) to a system administrator to warn of a significant broadcaster load. In the event that no non-VIP servers are available to a given broadcaster (because all non-VIP servers are at true maximum capacity), the process displays "No Available Server" as an error message on the display of the broadcaster client device.

V. Media Server Process

FIGS. **5**A through **5**C show a process flow **550**A, **550**B, **550**C, and **550**D illustrating a media server process for the RTMP and WebRTC media servers **320** and **360** shown in FIGS. **2** and **3**, according to one inventive implementation. These process flows include a "server monitor" process and a "video uploader" process that each of the RTMP and WebRTC media servers implements as they receive and process live streams from various broadcasters.

Regarding the "server monitor" process, a given media server periodically reports server statistics to be stored in the database **420**, and queries the database to obtain a list of broadcaster streams that have been assigned to, and are connected to, the media server. For newly connected streams, the media server validates the stream information (e.g., StreamID), with the database, and if the stream is valid the media server starts a live transcoding process to provide different resolution copies of the live stream (e.g., 720p, 360p and 240p transcoded copies); in the case of a WebRTC media server, the media server also transcodes the VP8/WebRTC live stream to H.264 before providing the different resolution transcoded copies. In some implementations, the media server employs hardware-accelerated transcoding of the broadcaster's live stream (e.g., via graphic card processing) to ensure low latency of viewed transcoded copies of the live stream. The media then starts recording the highest resolution transcoded copy (e.g., 720p in the illustrated example) to provide a "raw video" recording, and notifies the database that the live stream has started and is available for viewing. Thereafter, the media server queues a first screenshot (thumbnail) for the live stream in the asynchronous queue (e.g., see **850** in FIG. **3**) for processing by the control server **500** (see FIGS. **18**A and **18**B), and also queues push notifications to notify subscribers and followers of the broadcaster that the broadcaster is online with a live stream (e.g., by providing a StreamID to the followers/subscribers).

Thereafter, while the broadcaster continues to provide a live stream, and if there are any HLS viewers (discussed further below in connection with FIGS. **8** and **9**A through **9**D), the media server begins an HLS segmentation process to create and update an HLS file suite comprising an HLS playlist, HLS chunklists, and HLS file segments for each of the transcoded different resolution copies of the broadcaster's live stream. The media server process also periodically queues in the asynchronous queue (e.g., every five seconds or so) additional screenshots/thumbnails of the live stream. Once the broadcaster has ended the live stream, the media server process stops the recording of the highest resolution transcoded copy, sends out a system message on the chat/system event socket(s) corresponding to the broadcaster's live stream that the stream has ended, stops the live transcoding process, and stores the stream end time in the database **420**. The media server process then also queues the upload of the "raw video" recording (the recording of the highest resolution transcoded copy) to the media server upload queue.

The video uploader process shown in FIG. **5**A reads from the media server upload queue and, if there are any entries in the queue, uploads the corresponding raw video recording of the broadcaster's live stream to data storage **440** (e.g., Amazon S3) and stores the upload time to the database **420**. The video uploader process also may notify a third-party transcoding service (e.g., see the transcoding server pool **800** in FIG. **3**) to provide transcoded different resolution copies of the recorded video to facilitate adaptive bitrate replay for one or more viewers.

VI. Viewer Stream Source Selection

FIG. **6** is a block diagram illustrating the media sources **300** and the web server(s) **700** of the servers and memory

storage devices **1000** shown in FIGS. **2** and **3**, as well as the first and second broadcaster client devices **100**A and **100**B and one of the viewer client devices **200**A, to facilitate a discussion of the selective coupling of an example viewer client device to one of the media sources, according to some inventive implementations. In tandem with FIG. **6**, FIG. **7** is a process flow diagram illustrating a viewer stream source selection algorithm **702** according to one inventive implementation, which in some examples may be performed by the web server(s) **700**.

As depicted representationally in FIG. **6**, in one aspect the web server(s) **700** essentially serve as a controllable switch to couple the viewer client device **200**A to one of an RTMP media server **320**, the RTMP CDN **340** (which is communicatively coupled to the RTMP media server(s) in a virtual private network **330**), a WebRTC media server **360** and the HLS serve architecture **360** to receive a copy of broadcaster's live stream of digital content. In the example of FIG. **6**, the web server(s) **700** has facilitated a connection between the viewer client device **200**A and the RTMP CDN **340** (as shown by the dashed line in FIG. **6**). However, as discussed below, the web server(s) **700** may facilitate a connection between the viewer client device **200**A and any one of the media sources **300** based at least in part on a number of viewers already receiving copies of the broadcaster's live stream. In one implementation, the database **420** stores user profiles for broadcasters and viewers, in which the user profile may include a user type (e.g., registered user, anonymous user, subscriber of one or more broadcasters, VIP user, media professional or media member, etc.); in this instance, the web server(s) **700** may facilitate a connection between the viewer client device **200**A and one of the media servers **300** based at least in part on a type or status of a user of the viewer client device **200**A and/or the number of viewers already receiving copies of the live stream.

More specifically, as shown in the process of FIG. **7**, if the viewer client device sends a request to the web server(s) **700** to view a copy of a given broadcaster's live stream (e.g., based on a StreamID for the live stream that the viewer client device received in a push notification), and the web server(s) **700** determine that there are fewer than a first number (e.g., 10) of viewers already receiving copies of the live stream (e.g., based on a viewing count for the stream maintained in the database **420**), the web server(s) provide to the viewer client device an address to connect directly to one of the RTMP media servers **320** or one of the WebRTC media servers **360** that is processing the broadcaster's live stream (depending on whether the broadcaster client device is a mobile H.264 or web-based VP8 client device). Irrespective of the number of viewers, the web server(s) **700** also provide an address to the viewer client device to connect directly to one of the media servers if a user of the viewer client device is a VIP subscriber or media professional. If however the user is not a VIP subscriber or media professional, and there are more than a first number of viewers already receiving copies of the live stream, the web server(s) provide to the viewer client device an address to connect to one of the CDN servers of the RTMP CDN **340**. However, if all CDN servers of the RTMP CDN **340** are at their maximum capacity (e.g., as reflected in server statistics stored in the database), the web server(s) **700** provide an address to the viewer client device to connect to the HLS server architecture **360**.

VII. HTTP Live Streaming (HLS) Server Architecture

FIG. **8** is a block diagram showing additional details of the HLS server architecture **380** of the servers and memory storage devices **1000** shown in FIGS. **2**, **3** and **6**, according to some inventive implementations. FIGS. **9**A through **9**D show a process flow illustrating an HLS stream viewing process **902**A, **902**B, **902**C and **902**D performed by the HLS server architecture **380** shown in FIG. **8**, according to one inventive implementation. As some of the discussion of the HLS server architecture **380** relates to processing of a live stream at a media server, reference is made again to the media server process discussed above in connection with FIGS. **5**A, **5**B, and **5**C.

HTTP Live Streaming (HLS) is a conventional HTTP-based media streaming communications protocol, in which a live media stream (e.g., video and accompanying audio) is divided up or "segmented" by an HLS media server into a sequence of small files that may be downloaded to a viewer client device via HTTP communications with the HLS media server, wherein each downloaded file represents one short segment or "chunk" of a copy of the live stream. As respective chunks of the copy of the live stream are downloaded and played by the viewer client device, the client device may select from multiple different alternate streams containing the same video/audio material transcoded by the media server at a variety of data rates (e.g., at different resolutions), allowing the HLS streaming session to adapt to the available data bit rate/bandwidth of the client device's connection to the HLS server. HLS connections are, by definition, not persistent connections between the HLS media server and the viewer client device, since requests for and delivery of HLS content uses only standard HTTP transactions. This also allows HLS content to be delivered to multiple viewer client devices over widely available HTTP-based content delivery networks (CDNs).

With reference again to the media server process in FIGS. **5**A, **5**B, and **5**C, as a broadcaster's live stream is received by a media server it is cached for some amount of time (e.g., 10 to 30 seconds). The broadcaster's live stream typically includes a succession of frames at some frame rate (e.g., 30 frames/sec), and the succession of frames includes multiple "keyframes" associated with video encoding/compression. Such keyframes include the "full" content of an instant of the video, and these keyframes reset the basis of calculation (compression/estimation) for ensuing video information; in conventional video encoding/compression techniques, compressed frames between keyframes essentially include only information representing what has changed in the content between respective frames, and not the entire visual content for corresponding instants of the video. Increasing the frequency of keyframes in the stream of video frames reduces any errors that may be introduced in the compression process, as such errors would have a shorter lifespan (there would be fewer numbers of compressed frames between keyframes).

As indicated in FIG. **5**B, an incoming live stream from a broadcaster and received by a media server (e.g., incoming H.264 from an RTMP broadcaster client, or VP8 from a WebRTC broadcaster client that has been transcoded to H.264) is transcoded (e.g., by the media server) to provide different resolution copies of the live stream at corresponding different bitrates (e.g., to facilitate adaptive bitrate streaming, as noted above). For example, the broadcaster's live stream may be transcoded to provide 720p, 360p and 240p different resolution copies of the live stream. As part of the transcoding process, the media server may be configured such that the keyframe interval for each transcoded copy is a predetermined value, and the keyframe interval for the transcoded copies may be the same as or different than a keyframe interval associated with the broadcaster's incom-

ing live stream. Conventional examples of keyframe intervals that may be configured at a media server for transcoded copies of the live stream range from about 60 frames to 300 frames of video, and in some instances as high as 600 frames (at an exemplary frame rate of 30 frames/second, the associated time durations for such keyframe intervals range from two seconds for a keyframe interval of 60 frames to 10 seconds for a keyframe interval of 300 frames, and in some instances as high as 20 seconds for a keyframe interval of 600 frames).

As discussed above in connection with FIG. **5**C, to implement HLS each of these different resolution copies is divided into small segments of video based in part on the keyframe interval of the copies. More specifically, the media server may be configured to set a target segment length (duration) of each segment into which the transcoded copy of the live stream is divided. An example of a conventional target segment duration for HLS is 10 seconds; however, as discussed below, in some implementations the media server is particular configured to have a significantly lower target segment duration to facilitate the functionality of the HLS server architecture **380** in processing copies of segmented live streams.

With reference again to FIG. **5**C, the media server ultimately divides each copy of the live stream into respective video segments having a duration that is as close as possible to the target segment duration, with the proviso that a segment must start on and include a keyframe but may include one or more keyframes (i.e., the segment duration in practice is based on the target duration configured in the media server, some multiple of keyframes, and the frame rate of the transcoded copy). For purposes of illustration, and taking a conventional target segment duration of 10 seconds, a frame rate of 30 frames/second, and a keyframe interval of from 60 to 300 frames, each conventional 10 second HLS segment may have 1 keyframe (given a keyframe interval of 300 frames) or up to 5 keyframes (given a keyframe interval of 60 frames).

For each transcoded different resolution copy of the broadcaster's live stream, the HLS segments of the copy are stored as small files (referred to in HLS as .ts files). Thus, in an example in which there are 720p, 360p and 240p transcoded copies of the live stream, there are three sets of .ts files being generated and stored in memory at the media server as each of the copies are segmented by the media server. For each set of .ts files corresponding to a different resolution copy of the live stream, a "chunklist" is created and maintained by the media server that includes a list of pointers (e.g., relative URLs) to corresponding .ts files stored in memory; accordingly, in the example of three different resolution copies, there would be three different corresponding chunklists.

The number of pointers in a given chunklist may be referred to as the "HLS window" or "HLS buffer length," and this HLS window/buffer length may be set as a configuration parameter for the media server. One conventional example of an HLS window/buffer length is pointers to corresponding .ts files. The number of pointers in the chunklist multiplied by the duration of the HLS segment represented by each .ts file is referred to as the "HLS latency," because a viewing client that requests an HLS copy (i.e., succession of .ts files) typically does not start downloading a first .ts file representing a video segment until the chunklist is completely populated with the set number of pointers to corresponding .ts files (the HLS window/buffer length). Given the example above of a conventional target segment duration of seconds, this results in a conventional HLS latency on the order of 100 seconds. This HLS latency also may be viewed as a "buffer time" that provides for stability of the HLS stream in the event of communications issues or interruptions in network connectivity; the latency arising from the segment duration and HLS window/buffer length provides for the overall download and playback time of the .ts file segments before another chunklist is downloaded by a viewer client device, thereby mitigating potential connectivity issues that may occur between the client device and a CDN server during this buffer time (presuming that, under normal circumstances, it is quicker for the client to download a .ts file segment than it is for the client to play the segment). As new .ts files get created in the segmenting process for a given resolution copy of the live stream, the media server puts a new pointer to the newest .ts file into the corresponding chunklist and, once the chunklist is filled the first time with the set number of pointers corresponding to the buffer length, the oldest pointer gets "bumped out" of the chunklist when a new segment/pointer is generated, in a first-in-first-out (FIFO) manner.

Below is an example of a chunklist that includes six pointers to corresponding .ts files representing HLS video segments:

```
==> curl -v https://we109.media.castrlive/t1/ngrp:397965_all/chunklist_w1844413579_b2096000.m3u8
* Trying 198.204.252.202 . . . .
* TCP_NODELAY set
* Connected to we109.media.castr.live (198.204.252.202) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: *.media.castr.live
* Server certificate: Go Daddy Secure Certificate Authority—G2
* Server certificate: Go Daddy Root Certificate Authority—G2
> Get/t1/ngrp:397965_all/chunklist_w1844413579_b2096000.m3u8 HTTP/1.1
> Host: we109.media.castr.live
> User-Agent: curl/7.51.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Accept-Ranges: bytes
< Server: WowzaStreaming Engine/4.7.0.01
< Cache-Control: no-cache
< Access-Control-Allow-Origin: *
< Access-Control-allow-Credentials: true
< Access-Control-Allow-Methods: OPTIONS, GET, POST HEAD
< Access-Control-Allow-Headers: Content-Type, User-Agent, If-Modified-Since, Cache-Control, Range
< Date: Thu, 8 Jun 207 21:10:47 GMT
< Content-Type: application/vnd.apple.mpegurl
< Content-Length: 368
<
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION: 4
#EXT-X-MEDIA-SEQUENCE:349
#EXTINF:2.352,
media_w1844413579_b2096000_349.ts
#EXTINF:2.04,
media_w1844413579_b2096000_350.ts
#EXTINF:2.002,
media_w1844413579_b2096000_351.ts
```

#EXTINF:2.001,
media_w1844413579_b2096000_352.ts
#EXTINF:2.001,
media_w1844413579_b2096000_353.ts
#EXTINF:2.001,
media_w1844413579_b2096000_354.ts

In addition to a chunklist for every different resolution copy of the broadcaster's live stream, the media server also creates an HLS "playlist" file (e.g., having a file extension .m3u8) corresponding to the broadcaster's live stream. The HLS playlist includes a list of the transcoded different resolution copies of the live stream, and for each item in the list the playlist also includes a corresponding bandwidth/bitrate, one or more codecs for encoding the copy of the stream, and a pointer (e.g., relative address or URL) to the corresponding chunklist:

720p copy—bitrate—codec(s)—relative URL1 for chunklist1
360p copy—bitrate—codec(s)—relative URL2 for chunklist2
240p copy—bitrate—codec(s)—relative URL3 for chunklist3

An example of an HLS playlist file is provided below:

==> curl -v https://we109.media.castrlive/t1/ngrp:397965_all/playlist.m3u8
* Trying 198.204.252.202 . . . .
* TCP_NODELAY set
* Connected to we109.media.castr.live (198.204.252.202) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: *.media.castr.live
* Server certificate: Go Daddy Secure Certificate Authority—G2
* Server certificate: Go Daddy Root Certificate Authority—G2
> Get/t1/ngrp:397965_all/playlist.m3u8 HTTP/1.1
> Host: we109.media.castr.live
> User-Agent: curl/7.51.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Accept-Ranges: bytes
< Access-Control-Expose-Headers: Date, Server, Content-Type, Content-Length
< Server: WowzaStreaming Engine/4.7.0.01
< Cache-Control: no-cache
< Access-Control-Allow-Origin: *
< Access-Control-allow-Credentials: true
< Access-Control-Allow-Methods: OPTIONS, GET, POST, HEAD
< Access-Control-Allow-Headers: Content-Type, User-Agent, If-Modified-Since, Cache-Control, Range
< Date: Thu, 8 Jun 207 21:07:51 GMT
< Content-Type: application/vnd.apple.mpegurl
< Content-Length: 368
<
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-STREAM-INF:BANDWIDTH=2296000, CODECS="avc1.77.41,mp4a.40.2",RESOLUTION=1280×720
chunklist_w1844413579_b2096000.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1031000, CODECS="avc1.77.31,mp4a.40.2",RESOLUTION=640×360
chunklist_w1844413579_b946000.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=449000, CODECS="avc1.66.30,mp4a.40.2",RESOLUTION=426×240
chunklist_w1844413579_b414000.m3u8
* Curl_http_done: called premature==0
* Connection #0 to host we109.media.castr.live left intact

Thus, the HLS "file suite" corresponding to a broadcaster's live stream includes:

A playlist of different resolution copies with corresponding pointers to chunklists
The chunklists, each containing a set of pointers to corresponding .ts files
The .ts files pointed to in the chunklist for each different resolution copy

To play an HLS copy of a live stream, the viewer client device first requests a copy of the corresponding HLS playlist file from the media server. Based on the available bandwidth between the viewer client device and the media server at any given time, once the playlist is received the viewer client device selects the most appropriate resolution copy from the playlist having a bit rate that may be accommodated by the available bandwidth; this provides for adaptive bit rate streaming in that, from time to time, the viewer client device may select a different resolution/different bitrate copy of the live stream from the list of copies in the HLS playlist based on changes in the available bandwidth (e.g., quality of connection) between the viewer client device and the media server. Once the desired copy is selected from the playlist based on available bandwidth, the viewer client device then requests from the media server the current chunklist associated with the selected copy of the live stream, based on the corresponding pointer to the chunklist that is present in the playlist. As noted above, the chunklist for each copy of the live stream is continuously updated by the media server (FIFO) as new .ts files are created by the media server. Once the viewer client device retrieves the chunklist, it can then in turn begin retrieving the respective .ts files pointed to in the chunklist (e.g., via corresponding relative URLs) and playing the video segments represented in the .ts files. The viewer client device repeatedly requests the appropriate chunklist from the media server (e.g., after every video segment is played) to retrieve a current version of the chunklist. In the foregoing manner, as noted earlier, data/files are transmitted from the media server to the viewer client device upon request pursuant to HTTP, as opposed to streaming data continuously between the media server and the viewer client device via a persistent data connection.

Conventionally, for every request from a viewer that a media server receives for an HLS copy of a live stream, the media server creates a new HLS file suite for the requester, including an HLS playlist, associated chunklists, and sets of .ts files. Typically, such requests for an HLS copy of a live stream would arrive at the media server from respective (e.g., geographically distributed) servers of a CDN that are in turn communicating with respective (e.g., geographically distributed) viewer client devices. As HLS viewer demand increases for copies of a particular broadcaster's live stream, the load (e.g., CPU demand) on the media server increases based on the media server's process for generating a new HLS file suite for each new HLS requester.

Moreover, given that different viewer client devices may be requesting (via corresponding different CDN servers) an HLS copy of the live stream at different points in time, the Inventors have recognized and appreciated that significant synchronization issues arise amongst respective viewers based at least in part on the media server's process for

generating a new HLS file suite for each new request. More specifically, because the media server creates different HLS file suites at different times for different requesters, a first requester viewing a first copy of the live stream likely sees the video content some time earlier than or later than a second requester viewing a second copy of the live stream, because at any given time the respective requesters may be downloading and playing different video segments from their respective chunklists. For conventional HLS applications, this lack of synchronization amongst respective viewers typically would not pose any problems in viewer experience.

However, the Inventors have recognized and appreciated that in the example context of multiple viewers viewing respective copies of a broadcaster's live stream of video-based commentary regarding a live event, and also receiving and displaying event information as real-time updates about the event, this lack of synchronization amongst respective HLS viewers may significantly and adversely impact viewer experience. For example, particularly in the context of a "second screen experience," two different HLS viewers watching the same event on a first screen and watching the same broadcaster's live video-based commentary on a second screen may see the broadcaster's video-based commentary significantly out of synchronization with the live event on the first screen, and may receive and display event information (e.g., event score updates) on the second screen that are noticeably out of synchronization with the live event and/or the broadcaster's video-based commentary. Furthermore, if both of the viewers happen to be watching the same event together at the event venue on the same first screen (e.g., together in the same room at a gathering or party), they may find that their respective copies of the broadcaster's video-based commentary are noticeably out of synchronization on their respective viewer client devices.

In view of the foregoing technical problems relating to HLS viewer synchronization and media server loading, the Inventors have implemented an inventive technical solution via an HLS server architecture **380** that provides caching and amplifying functionality to address the above-noted technical problems. An example of such an HLS server architecture is shown in FIG. **8** and discussed in detail below, and FIGS. **9**A through **9**D illustrate flow diagrams that outline the process by which a given viewer client device requests and receives an HLS copy of a broadcaster's live stream via the HLS server architecture shown in FIG. **8**.

In considering the various HLS multiple-viewer synchronization issues that are addressed by the HLS server architecture shown in FIG. **8** and the processes outlined in FIGS. **9**A through **9**D, the Inventors also have considered and addressed the overall latency implications of conventional HLS stream delivery in light of the inventive HLS server architecture disclosed herein. To this end, the Inventors have considered unconventional settings (e.g., at the media server) for various parameters relating to HLS streams such as keyframe interval, target segment duration, and HLS window/buffer length for chunklists. Recall in the discussion above that conventional examples of these parameters respectively include a keyframe interval of from 60 to 300 frames, a target segment duration of 10 seconds, and an HLS window/buffer length of 10 .ts files or "chunks," giving rise to a conventional HLS latency on the order of 100 seconds. Such a latency is practically untenable in the example context of multiple viewers viewing the live event itself in person or on a first screen, viewing respective HLS copies of a broadcaster's live stream of video-based commentary regarding the live event as a second screen experience (which would be 100 seconds out of synchronization with the live event/first screen), and also receiving and displaying on the second screen event information as real-time updates about the event (which would be 100 seconds out of synchronization with the video-based commentary on the second screen).

The Inventors have recognized and appreciated that the above-mentioned parameters may be specifically selected (e.g., via configuration of the media server) to significantly reduce latency while sufficiently maintaining stability of HLS content delivery. To this end, in one example inventive implementation, the keyframe interval for transcoded copies of the live stream may be set to 30 frames (i.e., significantly fewer than 60 to 300 frames), the target video segment duration may be set to two seconds (i.e., significantly lower than 10 seconds, and such that the succession of HLS segments respectively have two keyframes each at a frame rate of 30 frames/second), and the HLS window/buffer length may be set to from four to six segments in a chunklist (as opposed to 10 chunks in a chunklist as suggested conventionally). These parameters result in a significantly reduced HLS latency of approximately 8 to 12 seconds, as compared to a conventional HLS latency on the order of 100 seconds

As shown in FIG. **8**, in one implementation an HLS caching and amplifying server architecture **380** includes a "mother" server **382** and may also include one or more "child" servers **384**A through **384**D, disposed between a media server and an HLS CDN server pool **388**, in which the HLS mother server acts as a single "virtual viewer" from a given media server's perspective. While FIG. **8** shows multiple child servers, it should be appreciated that in various inventive implementations the HLS server architecture need not have any child servers, or may only have one child server; however, the inclusion of one or more child servers in the inventive HLS server architecture facilitates enhanced scaling and reduced loading (e.g., CPU usage/bandwidth) on the mother server.

In example implementations, the HLS mother server, as well as one or more child servers, may be implemented as a customized NGINX-based caching server. Based on a single copy of an HLS file suite **375**A (e.g., single playlist, associated chunklist(s), and associated .ts file segments) for a given broadcaster's live stream as provided by a media server **320**/**360** and received by the mother server **382** of the HLS server architecture, the mother server caches and passes on copies **375**B of the elements of the file suite (as requested) to one or more child servers, which in turn cache and pass on copies **375**C of the elements of the file suite to one or more geographically-distributed servers of a conventional (e.g., global) CDN (serving as an HLS CDN in tandem with the mother-child server architecture). In this manner, the mother and child servers of the HLS architecture act as caching and amplifying servers, so that identical HLS streams may be served from the HLS CDN server pool to multiple viewers of a given broadcast in a significantly narrower synchronization window than conventionally possible. In particular, in one example implementation, all HLS viewers receiving a copy of a broadcaster's live stream via the HLS server architecture shown in FIG. **8** are at most less than one HLS file segment duration out of synchronization with each other (referred to herein as "viewer segment concurrency").

As noted above, in conventional HLS, a viewer client device does not maintain a persistent connection with an HLS media server; similarly, by default, HLS media servers do not allow caching of HLS files (e.g., playlists, chunklists

and .ts files). In particular, as illustrated above in the examples of a conventional HLS chunklist and a playlist, these files respectively include an explicit instruction that prevents caching (i.e., "Cache-control: no-cache"). For different types of files, cache-control conventionally may be set for some time period that allows a file to be temporarily stored (i.e., cached) by a requesting server, after which a fresh copy of the file needs to be requested from its origin server by the requesting server; as noted above, however, caching is conventionally prohibited for HLS files by an explicit instruction in the files.

Unlike conventional HLS, in inventive implementations of the HLS server architecture shown in FIG. **8**, when a first requester requests a copy of a given broadcaster's live stream the HLS mother server establishes and maintains a persistent connection to the media server (e.g., the RTMP or WebRTC media server receiving the broadcaster's incoming live stream). In this manner, as long as the broadcaster is generating the live stream, at least one requester is requesting a copy of the live stream, and no matter how many requests may be made by globally-distributed CDN servers for copies of the live stream on behalf of requesting viewer client devices, the media server only sees the load of one requester (i.e., the HLS mother server). In this capacity, the HLS media server does not have to make copies of the HLS file suite to provide for additional requesters of the broadcaster's live stream as would be required in conventional HLS; instead, the HLS mother server requests and receives a single copy of the playlist file from the media server. As discussed further below in connection with FIGS. **9**A through **9**D, the HLS mother server requests the single copy of the playlist file from the media server in response to a request for the playlist file made by one of the HLS child servers to the mother server. The HLS child server makes such a request to the mother server in response to a request for the playlist file made by a CDN server to the child server on behalf of a requesting viewer client device. In a manner similar to that noted above, an HLS child also may open up and maintain a persistent connection with the HLS mother.

In an example implementation, when the HLS mother server requests and receives the HLS playlist file from the media server, the HLS mother server re-writes the caching rule in the received playlist file to allow the playlist to be cached for some period of time for which a broadcaster may be expected to provide the live stream (e.g., some number of hours up to 24 hours or 86,400 seconds); in particular, the HLS mother server strips the "Cache-control: no-cache" setting from the received playlist file and replaces it with a new cache-control command having some duration of caching time. The HLS mother server then caches the revised playlist file (for the duration of the new caching time) and typically the playlist file need not be requested again from the media server. A copy of this revised playlist file with a re-written caching rule in turn is provided upon request to one or more of the HLS child servers, which in turn cached the revised playlist file and pass additional copies of the revised playlist file to one or more CDN servers so that the playlist file is ultimately provided to one or more requesting viewer client devices. Based on the re-written caching rule, each of the involved servers may store a copy of the revised playlist file for the duration of the broadcaster's live stream and need not request it again; and again, as noted above, the media server only "sees" one requesting viewer and provides one playlist, no matter how many actual viewers may be requesting a copy of the broadcaster's live stream.

More specifically, as shown in FIG. **9**A, when a given viewing client device wishes to receive a copy of a broadcaster's live stream, the client device first queries a CDN server for a copy of the HLS playlist file corresponding to the broadcaster's live stream. If the CDN server has a copy of the playlist (e.g., based on a previous request from another viewer client device), the CDN server returns the playlist to the currently requesting client device. If however the CDN server does not have a copy of the revised playlist, the CDN server connects to a CDN load balancer **386** and in turn requests a copy of the revised playlist from one of the HLS child servers as determined by the load balancer.

If the HLS child server has a copy of the revised playlist (e.g., based on a previous request from a CDN server), the HLS child server returns the revised playlist to the currently requesting CDN server (which in turn passes the playlist on to the requesting viewer client device). If however the HLS child server does not have a copy of the revised playlist, the HLS child server requests a copy of the revised playlist from the HLS mother server.

If the HLS mother has a copy of the revised playlist (e.g., based on a previous request from one of the HLS child servers), the HLS mother server returns the revised playlist to the currently requesting HLS child server. If however the HLS mother server does not have a copy of the playlist (e.g., because this is the first request for a copy of the broadcaster's live stream), the HLS mother server establishes a persistent connection with the appropriate media server (e.g., based on the relative URL for the HLS copy of the stream at a given media server), requests a copy of the playlist, and re-writes the caching rule for the playlist as discussed above. The HLS mother then caches the revised playlist, returns the revised playlist to the currently requesting HLS child server. The child server in turn caches the revised playlist and passes the revised playlist on to the requesting CDN server, which in turn also caches the revised playlist and passes the revised playlist on to the requesting viewer client device.

As shown in FIGS. **9**A through **9**D, once the viewer client device has the playlist, it selects from the playlist the appropriate resolution copy of the live stream based on the associated bitrate of the copy and the available bandwidth between the viewer client device and the CDN server. Based on the selected copy of the live stream, the viewer client device then requests from the CDN server the corresponding chunklist. In a manner similar to the request for the HLS playlist, each of the CDN server, an HLS child server, and the HLS mother server may be queried in turn for a copy of the corresponding chunklist.

However, an important distinction between the playlist and a requested chunklist relates to the "freshness" of the chunklist and the re-writing of the chunklist's caching rule by the HLS mother server. In particular, whenever the HLS mother server requests a given chunklist from the media server, the mother server re-writes the caching rule in the received chunklist file to allow the chunklist to be cached for some period of time, for example, the segment duration corresponding to a single .ts file (e.g., two seconds). In particular, the HLS mother server strips the "Cache-control: no-cache" setting from the chunklist file and replaces it with a new cache-control command having some duration of caching time (e.g., corresponding to a segment duration). In one aspect, a caching time corresponding to a segment duration is contemplated given that the chunklist does not change during this duration (and thus, any requests for the chunklist during this duration are generally unnecessary). The HLS mother server then caches the revised chunklist file (for the duration of the new caching time) and a copy of this revised chunklist file with a re-written caching rule in turn

is provided upon request to one of the HLS child servers, which in turn also caches the revised chunklist and passes a copy of the revised chunklist file to a CDN server so that the chunklist file is ultimately provided to the requesting viewer client devices. Based on the re-written caching rule, each of the involved servers may cache a copy of the updated chunklist file for up to but no more than the specified caching time, which ensures that each copy of the chunklist stored on a given server is "fresh" (e.g., within one segment duration) for downloading to the requesting viewer client device, while also mitigating unnecessary resources spent on attending to requests for chunklists during a time period in which there are no changes to the chunklist. In an alternate implementation, a given child server may again re-write the caching rule for a chunklist file to prevent caching of the chunklist by a requesting CDN server (and thereby cause the CDN server to request the chunklist from the child server every time the chunklist is requested from the CDN server by a viewer client device, even if respective requests come from one or more viewer client devices within a segment duration).

Referring again to FIGS. **9**A through **9**D, and considering a non-limiting example implementation in which the segment duration corresponding to a .ts file is two seconds and the CDN servers maintain the same revised caching rules as the HLS mother and child servers, FIGS. **9**A through **9**D illustrates that when a requesting viewer client device does not have a chunklist, it requests the chunklist from a CDN server. If the CDN server does not have the chunklist, or if the chunklist cached on the CDN server is more than two seconds old (i.e., exceeds the cache time), the CDN server requests the chunklist from an HLS child server; otherwise, the CDN server returns a "fresh copy" of the chunklist to the requesting client. A similar process is repeated for the HLS child server and the HLS mother server, i.e., if the HLS child server does not have the chunklist, or if the chunklist cached on the child server is more than two second old, the child server requests the chunklist from the mother server; otherwise the child server returns a fresh copy of the chunklist to the requesting CDN server. If the HLS mother server does not have the chunklist, or if the chunklist cached on the mother server is more than two seconds old, the mother server requests the chunklist from the media server, re-writes the caching rule in the chunklist file, caches the revised chunklist file, and returns a fresh copy of the chunklist to the requesting child server (which in turn passes the fresh copy of the chunklist to the requesting CDN server and the requesting client device).

Once the requesting viewer client device has a fresh copy of the chunklist, the viewer client device begins requesting the respective .ts files or "chunks" pointed to in the chunklist. In some respects, as shown in FIGS. **9**A through **9**D, this process is similar to the processes outlined above for requesting the playlist and requesting one of the chunklists pointed to in the playlist. For example, the requesting viewer client device requests a chunk from a CDN server and, if the CDN server has the requested chunk (e.g., because another requesting viewer previously requested the same chunk from the same CDN server and the CDN server already has the chunk cached), the CDN server returns the chunk to the client device for playing the video segment represented in the chunk. If however the CDN server does not have the chunk cached, it requests the chunk from an HLS child server (e.g., via the CDN load balancer). A similar process is repeated for the HLS child server and the HLS mother server. If ultimately the mother server does not have the chunk cached and needs to request the chunk from the media server (e.g., because this is the first viewer request for this chunk), the mother server requests the chunk from the media server, re-writes the caching rule in the chunk file (e.g., to change the caching rule from "no-cache" to some period of time, for example one hour), caches the revised chunk, and returns a copy of the chunk to the requesting child server (which in turn passes the copy of the chunk to the requesting CDN server and the requesting client device).

Once the viewer client device has downloaded all chunks pointed to in the chunklist, it plays them in turn, deletes the current copy of the chunklist that the viewer client device has cached, and then again determines the appropriate resolution copy of the live stream to request based on the associated bitrates of the different resolution copies and the available bandwidth between the viewer client device and the CDN server. Typically, it takes less time for a client to download a chunk then to play it; accordingly, if there are network issues, the copy of the stream can keep playing on the viewer client device while it downloads new chunks. For example, if the client successfully downloaded three chunks (six seconds of video) in two seconds of wall clock time, there remains a four second buffer of video at the client device in case the fourth chunk has a delay in retrieval.

The foregoing process of requesting and receiving an appropriate fresh chunklist based on available bandwidth, and downloading and playing the chunks pointed to in the chunklist, is repeated for the duration of the broadcaster's live stream. For example, if the media server stops receiving the broadcaster's live stream, the media server may provide a message to the HLS mother server (e.g., in response to a request from the mother server for a fresh chunklist) that the live stream has been terminated; alternatively, the media server may provide an empty chunklist to the HLS media server, which essentially would ultimately terminate the iterative requesting process and the connection between the media server and the mother server would time out.

In other aspects, the HLS mother server shown in FIG. **8** monitors the current pool of media servers that may be servicing different broadcasters' live streams (e.g., as indicated in the database of the servers/memory storage devices **1000**), and self-configures to provide for custom routing (e.g., via relative URLs) between a requesting CDN server and a particular media server to appropriately retrieve a requested HLS copy of a given broadcaster's live stream (i.e., via the appropriate playlist and associated chunklists and .ts files). For example, custom routing functionality of the mother server may allow the targeting of specific media servers via a single entry URL (e.g., https://hls.media.castr.live/we90/t1/ngrp:123456_all/playlist.m3u8 requests retrieval of the adaptive HLS playlist from server "we90" for stream 123456, which the mother server internally translates to https://we90.media.castr.live/t1/mgrp:123456_all/playlist.m3u8 and thereby requests the playlist from the appropriate server, for which, when received, the mother server re-writes the caching rule, caches the revised playlist, and passes on the revised playlist to a requesting child server).

As noted earlier, in some implementations the HLS CDN shown in FIG. **8** that makes requests to one or more HLS child servers may be provided as the Amazon Cloudfront CDN. In any event, the geographically-distributed servers of the CDN cache to the various elements of the HLS file suite and can serve these from a variety of geographic locations to provide a virtually infinite number of HLS viewers using only a relatively small HLS CDN pool; and, irrespective of the number of CDN servers requesting content on behalf of

respective viewers, the CDN serves the content quickly, and the media server sees only a single virtual viewer as the HLS mother server. In one aspect, the different "layers" of servers in the HLS server architecture introduce some degree of latency between a given broadcaster's live stream and the viewer client devices; however, as noted above, all viewer client devices have "viewer segment concurrency," and the overall average latency for all viewers is nonetheless significantly reduced (e.g., as compared to conventional HLS). For example, given an example chunk segment duration of two seconds, and an example HLS window/buffer length of four segments, there may be up to eight seconds of latency introduced by the HLS segmenting process and another approximately two seconds of latency introduced by the transfer of files through the HLS server architecture. It should be appreciated that the various concepts discussed herein relating to the HLS server architecture are similarly applicable to other segmented live video streaming protocols (e.g., MSS, HDS, MPEG-DASH) for which inventive server architectures are contemplated by the present disclosure.

VIII. Control Server and Associated Services/Processes

FIG. **10** illustrates some of the functionality (e.g., services and other processes) performed by the control server **500** shown in FIGS. **2** and **3**, according to one inventive implementation. As noted above, the control server **500** is coupled to the memory system **400**, one or more event information providers **55**, one or more news feeds (RSS) **65** or other news sources, and the socket server(s) **600**. In one aspect, the control server **500** periodically retrieves various event information from the event information provider **55** and/or news from the news feeds **65** that is germane to respective broadcasters' video-based commentary. In another aspect, the control system **500** may store at least some portion of retrieved event information and/or news in the memory system **400**. More generally, the control server **500** implements a number of services/processes that govern functionality of other servers and devices in the servers/memory storage devices **1000**; examples of such control system services/processes include, but are not limited to: an RTMP media server scaling process to add or remove servers from the one or more RTMP media servers **320** of the media sources **300** (see FIG. **11**); an RTMP CDN server scaling process to add or remove servers from the RTMP CDN **340** of the media sources **300** (see FIG. **12**); a live stream and media server watchdog process (see FIGS. **13-14**); an event data ingress process (see FIG. **15**); a live event data monitor process (see FIG. **16**); an asynchronous task processor (see FIG. **17**); and a live stream thumbnail/screenshot acquisition process (see FIG. **18**).

1) Server Auto-Scaling Systems and Watchdogs

FIGS. **11**A through **11**C show a process flow diagram illustrating an RTMP media server scaling system service method **1102**A, **1102**B and **1102**C performed by the control server of FIG. **10**, according to one inventive implementation. In the method shown in these figures, the control servers automatically scale the number of RTMP media servers **320** of the media sources that are available for broadcasters based in part on the capacity demand for the servers (e.g., number of broadcasters providing live streams). The control server monitors various media server statistics that are maintained in the database **420** (e.g., number of active servers in the RTMP media server pool; servers marked for shutdown; individual server information such as server status active/shutdown, numbers of active connections to live streams, current capacity, date/time of when server first came online for availability, etc.) and brings servers in and out of the RTMP media server pool based at least in part on the server statistics. In various aspects, the control server maintains a minimum number of servers (e.g., at least two, or a minimum capacity corresponding to approximately double the cumulative traffic at a particular time) in the RTMP media server pool to allow for spikes in stream creation, and also provides for various buffering times to allow new servers to come online. FIGS. **12**A through **12**C show a process flow diagram illustrating an RTMP CDN server scaling system service method **1202**A, **1202**B, and **1202**C performed by the control server of FIG. **10**, according to one inventive implementation, that is similar in many respects to the method **1102**A, **1102**B and **1102**C performed for the media server scaling service.

FIGS. **13**A and **13**B show a process flow diagram illustrating a stream and server watchdog service method **1302**A, **1302**B performed by the control server of FIG. **10**, according to one inventive implementation. The stream watchdog performed by the control server essentially ensures that new streams created by broadcasters are valid and deletes streams that were created but not started, or that have been inactive for some period of time (e.g., 30 seconds). When streams are ended, the method generates final viewer statistics (e.g., stream duration, average number of viewers, maximum number of viewers, number of simultaneous viewers, viewers added, viewers left, etc.), broadcasts a "stream ended" system event message to the chat/system event socket(s) of the socket server(s) dedicated to the broadcaster's live stream, ends the recording of the live stream by the media server, and queues the recording to the video uploader queue of the media server process. The server watchdog portion of the method **1302**A, **1302**B monitors the RTMP media servers and the servers of the RTMP CDN and invokes the check RTMP Media/CDN server method **1402**A, **1402**B shown in FIGS. **14**A and **14**B. As part of the server watchdog process, for new servers the control server determines a capacity of the server (e.g., based on server type), and updates the database **420** with the capacity of respective servers, server class, launch time, status update (e.g., active and available for connections) and determines a total user/streamer capacity based on newly added servers. For servers that are already online, the server watchdog ensures that servers remain active for certain intervals (e.g., second intervals), automatically removes inactive servers from the pool, and reports active server status back to the database. If servers are marked for shutdown, the server watchdog archives server statistics, removes the server from the active server list stored in the database, and determines an updated total user/streamer capacity based on the removal of the server from the active list.

2) Event Information Ingress and Live Event Monitoring

In some inventive implementations, another significant role of the control server **500** shown in FIGS. **2**, **3** and **10** relates to collecting of event information and/or news (e.g., from external Internet providers), maintaining relevant event information and/or news in the database **420** (e.g., to facilitate selection of broadcasters to follow, and/or particular broadcaster live streams to view), and distributing the collected information to multiple broadcaster and viewer client devices in a relatively low-latency and synchronized manner with respect to broadcasters' video-based commentary.

In some implementations, the technical challenge of displaying event information and updates to same in a synchronized and low-latency manner amongst multiple client

devices is addressed in part by using a single control server **500** to gather and parse live event information captured in real-time. For example, for sporting events, game information may be obtained by the single control server from a dedicated third-party provider (e.g., STATS LLC). This single point of entry of event information prevents synchronization errors inherent in network communications. Once a change in event status has been detected (e.g., if a play clock updates), the control server provides these changes to the one or more sockets dedicated to the event (to which all viewers and broadcasters of video-based commentary regarding the event are communicatively coupled), resulting in a single synchronized update to all client devices and thereby significantly mitigating client-by-client latency and/or synchronization issues.

In some example implementations, the control server **500** implements two service methods relating to event information, namely, an event data ingress service and a live event data monitor service. The event data ingress service is performed with a first periodicity (e.g., once or twice a day) to maintain and update an event list in the database **420**. The live event data monitor service is performed with a second and more frequent periodicity (e.g., once a minute) to check for any events that are in progress and, if found, to retrieve fresh data about an in-progress event from the event information provider (e.g., at an even greater frequency, for example once a second). Similar services may be implemented by the control server **500** to ingest news on particular topics, trending threads, etc.

FIG. **15** shows a process flow diagram illustrating an event data ingress service method **1502** performed by the control server of FIG. **10**, according to one inventive implementation, and FIGS. **16**A and **16**B show a process flow diagram illustrating a live event data monitor service method **1602**A, **1602**B performed by the control server of FIG. **10**, according to one inventive implementation. In these methods, an event information provider is contemplated as supporting multiple different types of events for furnishing information (various types of sporting events such as basketball, football, baseball, hockey, etc.), and providing information for each instance of an event of a given event type (e.g., information for each of multiple basketball games, each of multiple football games, each of multiple baseball games).

For each event, the control server retrieves the raw information provided by the event information provider, and in some instances converts and/or compresses the raw information to provide a standardized format of essential data elements for storing in the database **420** and/or distribution to client devices (e.g., via broadcast of event messages having the standardized format to one or more dedicated sockets of the socket server(s) **600**). Examples of data elements for event information include, but are not limited to, a type of the event, an identifier for the event (EventID), a status of the event (e.g., pre-game, in-progress, final), score information for the event, team information for the event, a progress indicator or progress details for the event (e.g., quarter, period, inning, half-time; for baseball—balls, strikes, base information; for football—possession, down, yards to go; for basketball—timeouts, fouls), an event date and/or time of the event (e.g., actual or elapsed time information), and event participant data regarding participants in the event. In some examples, the control server further normalizes the event date and/or time to a particular reference frame (e.g., converting from UTC to EST/EDT).

In the process **1602**A and **1602**B shown in FIGS. **16**A and **16**B, the control server particularly queries the event information provider for a list of all events in a particular window around the current time (e.g., a 48 hour window, for events with start times 24 hours in the past through 24 hours in the future), to allow tracking of in-progress events (or identify any events that had inconsistent or incorrect start times or late modifications to event information). For each in-progress event, an event clock and other event information (e.g., score information, other more detailed information about the event) are updated frequently (e.g., once a second) to provide regular updates of event information messages that are broadcast to one or more dedicated event information sockets of the socket server(s) **600**.

3) Asynchronous Task Processing

FIGS. **17**A and **17**B show a process flow diagram illustrating an asynchronous task service method **1702**A, **1702**B performed by the control server of FIG. **10**, according to one inventive implementation. The control server periodically reads a task or task bundle from the asynchronous queue to initiate various other actions or processes in connection with the serves and memory storage devices **1000**. A number of different asynchronous system events may be implemented by this process, only some examples of which are illustrated in FIGS. **17**A and **17**B. For example, if an entry in the queue relates to a "Stream Started" system event, the asynchronous task processing sends out push notifications (including a StreamID) to followers and subscribers of the stream's broadcaster. Another system event processed by the asynchronous task process is when there is a new follower of a broadcaster's stream ("newFollowingStream"), for which the process loads user data and stream data, and attends to various user notifications as appropriate (e.g., email notifications, web push notifications). The asynchronous task processor is also responsible, in some implementations, for taking periodic screenshots/thumbnails of a live stream (as discussed below in connection with FIGS. **18**A and **18**B).

With respect to various push notifications handled by the control server **500** and/or the web server(s) **700** (or other servers of the architecture **1000**), it should be appreciated that specific apps on mobile client devices need not be open for a push notification to be received on the client device. Thus the client device may receive and display social media or text message alerts even when the device's screen is locked, and/or when the app receiving the push notification is closed. For iOS devices, for example, the Apple Push Notification Service API may be employed to enable the client app **5000** to receive various push notifications.

With reference again to FIG. **10**, the async queue monitoring is an application that runs on the control server and that looks at the current size of the asynchronous queue and will notify an administrator. Typically, the queue of tasks to process is small (e.g., at any given second it may be between 0-10 items), and if the queue grows to a larger size (e.g., 1000 items) the async queue monitor indicates to a system administrator that there is a problem in the asynchronous task processing (e.g., additional processing resources are required, or a looping event is getting processed and re-added to the queue instead of being removed).

4) Acquiring Screenshots/Thumbnails

FIGS. **18**A and **18**B show a process flow diagram illustrating a process **1802**A, **1802**B for taking a screenshot (thumbnail) of a live stream, performed by the control server of FIG. **10**, according to one inventive implementation (in other implementations, the web server(s) **700** of other servers of the architecture **1000** may perform the process of taking thumbnails of live streams pursuant to the general technique outlined in FIGS. **18**A and **18**B).

With reference again to FIG. **5**C and the media server process, the media server process queues to the asynchronous queue a first screenshot for a new live stream, and periodic updates to screenshots (e.g., every five seconds or so) during the duration of the live stream. These screenshot tasks are read by the asynchronous task process **1702**A and **1702**B discussed above in connection with FIGS. **17**A and **17**B and implemented by the process shown in FIGS. **18**A and **18**B.

In the process **1802**A, **1802**B, in one implementation screenshots are taken based on a broadcaster's live stream in H.264 (or transcoded to H.284 if the live stream is VP8/WebRTC from a web broadcaster). Screenshots are taken on the next available keyframe after the process is invoked. If the screenshot is not the first one taken, the stream information (e.g., in the database **420**) is updated with information relating to the newest screenshot, and the screenshot is added to archived screenshots (e.g., in the data storage **440**). The screenshot is then broadcast to the chat/system event socket of the socket server(s) **600** dedicated to the broadcaster's live stream.

Whenever a screenshot is taken of the broadcaster's live stream (particularly if it is the first screenshot), it may be resized for social media network requirements, and overlaid with graphics, watermarks, or promotional material. If the broadcaster requested for social share in creating the new stream (see discussion below regarding creation of new broadcaster streams), the process submits a link to the resized screenshot (e.g., an address or URL) to the indicated social network platform (e.g., Facebook, Instagram, Twitter, etc.), in some instances together with a "share graphic." In any case, the process determines the list of users that follow/subscribe to the broadcaster, and queues a system event message (e.g., "new FollowingStream") event for each subscriber to broadcast the first screenshot of the new live stream. As above, the stream information (e.g., in the database **420**) is updated with information relating to the screenshot, and the screenshot is archived (e.g., in the data storage **440**) and broadcast to the chat/system event socket of the socket server(s) **600** dedicated to the broadcaster's live stream.

With respect to sharing screenshots with social networks if elected by the broadcaster, in another implementation (not shown in FIGS. **18**A and **18**B), all screenshots of the broadcaster's live stream that are taken as of a given time are processed by a facial recognition algorithm to provide one of multiple options (e.g., the best of three screenshots) for selection by the broadcaster. For example, the process acquires a screenshot at 1, 3 and 5 seconds, and then every 5 seconds thereafter. The facial recognition algorithm detects candidate screenshots on rolling basis based on, for example, the clarity of the image, the quality of the face that is visible, and if the user is smiling. More specifically, every acquired screenshot is analyzed and then the "best three" are selected and presented as options to the broadcaster/viewer during social share. The broadcaster/viewer selects their preferred image, and the social share endpoint that is ultimately provided by the process to the selected social media platform(s) includes a link (e.g., address or URL) to the screenshot selected by the broadcaster.

IX. Client-Side Features (e.g., Functionality of the Client App)

Having provided various details of the servers and memory storage devices **1000** shown in FIGS. **2** and **3**, attention now turns to the functionality of the client devices relating to establishing user profiles (e.g., upon login), creating broadcaster stream sessions and providing live streams from broadcaster client devices to a media server, receiving copies of a live stream at a viewer client device (e.g., from a media server, the RTMP CDN, or the HLS server architecture), providing special effects graphics and animations (including animated real-time "scorebugs") on displays of client devices, and replaying copies of a recorded live stream from a broadcaster.

As noted earlier, unlike conventional scorebugs, screen animations, and/or other special effects graphics that are hard-embedded into live streams of a sports broadcast, in various inventive implementations disclosed herein graphics and effects are generated by the client device itself, separate from a given broadcaster's video-based commentary, and then integrated with (e.g., superimposed or overlaid on) the broadcaster's video-based commentary when rendered on the display of the client device. For mobile client devices, the client app **5000** executing on the device is particularly configured to render a variety of "studio-quality" graphics while nonetheless maintaining a small file size for the client app (e.g., less than 100 megabytes, and in some instances from approximately 60-70 megabytes); this allows the modestly-sized client app to be readily downloaded to a client device via a cellular network. In other aspects, client-side rendering of screen animations and/or other special effects graphics allows such animations and graphics to be user-interactive and/or user-customizable.

FIGS. **19**A and **19**B show a process flow diagram illustrating a user login process according to one inventive implementation, which in some examples may be performed by a client device and facilitated by one or more web servers **700** shown in FIGS. **2** and **3**. As illustrated, a login process may be implemented by phone (via SMS message with code sent to phone, and code validation), of via a social media network platform login process (e.g., Facebook, Twitter, Instagram). For new user accounts, a user may establish a user profile that is stored in the database **420** and that may be referenced by a UserID after creation, and include a user name, profile picture, and a user status or user "type" for the user (e.g., a VIP user or member, a media professional or member of the media).

1) Broadcaster Processes

FIGS. **20**A and **20**B show a process flow diagram illustrating a mobile broadcaster stream create process according to one inventive implementation, which in some examples may be performed by a broadcaster client device (pursuant to execution of the client app **5000**) and facilitated by one or more web servers (**700**) shown in FIGS. **2** and **3**. While much of the discussion above relates to an example in which a broadcaster wishes to provide a live stream of digital content including video-based commentary about a particular event, in other implementations the broadcaster may desire to create a live stream about a particular topic of interest (e.g., "anything"), or a news story, for example. For each of these options, the broadcaster may enter a title for the live stream, and the client device may request (e.g., from the web server(s) **700**) a list of events or news items for selection by the broadcaster, as well as a pre-populated list of tags (as noted above, event information and/or news may be ingressed by the control server **500**, and some event information and/or news may already be cached in the data cache **460** or stored in the database **420**).

The broadcaster may also enter tags to be associated with the live stream to facilitate searching and various social media functionality (e.g., to allow other users to search for and find the live stream based on various criteria represented

by the tags). The broadcaster may also elect other options in the stream creation process, examples of which include, but are not limited to, sharing an announcement of the stream starting on a social network platform, and enabling sharing of their location to other users (e.g., to facilitate viewing of the broadcaster's live stream by viewers based on the location of the broadcaster).

The broadcaster stream create process then submits a "stream create" request to the web server(s) **700**. If the broadcaster selected a particular event from the list of events about which to broadcast, an EventID associated with the event is included in the stream create request. Other contents of the stream create request includes, but is not limited to, an API key (to authenticate the user), the title of the stream, any tags selected, newsID (if news was selected), the broadcasters social network sharing options, and broadcaster location data (if permitted by the broadcaster). The web server(s) **700** in turn validates the API key, assigns a StreamID to the newly created live stream, runs the broadcast media server selection algorithm (e.g., see FIGS. **4**A and **4**B) to select a media server to which the broadcaster client device connects, and returns to the broadcaster client device the StreamID and the host name ("hostname") for the selected media server. The web server(s) **700** store in the database **420** a variety of stream information for the new live stream, which may include, but is not limited to, the StreamID, the UserID, the EventID, the DBshard, type of stream (RTMP/WebRTC), create time, hostname, title, tags, social notify options and social media platforms, location share option, location (if selected as an option) and, if the stream is associated with an EventID, an archived copy of event information at the stream create time.

FIGS. **21**A, **21**B, **21**C, **21**D, and **21**E show a process flow illustrating a mobile broadcaster active stream process **2102**A, **2102**B, **2102**C, **2102**D and **2102**E according to one inventive implementation, which in some examples may be performed at least in part by a broadcaster client device. In particular, the broadcaster client device accesses the media server selected by the web server(s) **700** via a particular URL (e.g., including the hostname in a path of the URL), as discussed below in connection with FIGS. **21**A through **21**E. The broadcaster client device then connects to a particular socket of the socket servers dedicated to the broadcaster's live stream, based in part on the StreamID provided by the web server(s), to establish a chat/system event channel. As noted above, in one aspect connections between client devices and a particular socket of a socket server are persistent authenticated connections, so that the number of users (broadcasters and viewers) connected to a particular socket (e.g., and currently watching a particular live stream and/or particular event) may be tracked. If the broadcaster's live stream is about an event, the broadcaster's client device also connects to a particular socket of the socket servers dedicated to the event, based on the EventID, to establish an event information channel.

In a "main loop" of the broadcaster client device stream active process (which for mobile clients is executed by the client app **5000**), an internal frame and time clock is periodically updated, and is used for animations and special effects graphics and synchronizing of some system messages that are received via the chat/system event socket (e.g., a default chat message displayed on the client device at the beginning of each new stream that says "keep it family friendly!"). The client device then checks to see if any further system messages or chat messages are received on the chat/system event channel, and displays chat messages and/or takes other actions in response to system messages such as "member added" (increase viewing count), "member_removed" (decrease viewing count), "new follower" (add notice to chat). Although only three system messages and corresponding actions are shown in FIG. **21**B, it should be appreciated that additional and/or other types of system messages may be received on the chat/system event channel (e.g., relating to other social networking functionality, and/or digital gifts) and initiate corresponding actions as part of the stream active process.

The client device next checks to see if any event messages or data is received on the event information channel (e.g., updates to event status, event score information, event clock, other event information). The client device then captures a camera frame for the live stream and sends the frame to the media server. The client device then checks the internal frame and time clock to see if any updates are needed to animations or special effects graphics (e.g., scorebugs) to be rendered on the display of the client device ("graphics/animation layers"). In some implementations, graphics and animations are updated at a rate of between 15-25 frames/second based on the internal frame and time clock. As noted above, in some implementations for mobile client devices, animated graphics and special effects are hard-coded in the client app as a series of individual frames (still-frame images), and rendered on the display in a "stop-motion" style according to the internal frame and time clock.

In the stream active process shown in FIG. **21**C, the process further queries for broadcaster input, examples of which include a request to end the stream, a request to share the stream, a request to view a list of viewers of copies of the live stream, interaction with the graphics/animations (e.g., "bottom third"), and a request to flip the camera. As also noted above, rendering graphics and animation layers on the client-side provides for user-interaction with the displayed graphics and animation layers. While not shown explicitly in FIG. **21**C, as discussed above interactions with graphics/animations ("set animation state to transition to open") may in some implementations launch a variety of other processes including, but not limited to, launching further graphics or animations, receiving additional information about the live sporting event (e.g., by thumbing-over a scorebug), or navigating to another Internet location to receive additional information relating to a live event.

In FIG. **21**D, the stream active process then queries if the stream state is set to close (e.g., in response to a broadcaster's request to end the stream, discussed immediately above). If not, the process returns to updating the internal frame and time clock. If the stream state is set to close, the client device disconnects from the media server, requests final stream statistics from the chat/system event channel, and displays an end of stream screen on the display of the client device.

FIGS. **22**A and **22**B show a communication flow diagram illustrating process flow elements and the server and/or memory storage devices involved in the communication flow for the processes shown in FIGS. **20**A and **20**B, and FIGS. **21**A-**21**E, as well as the media server processes shown in FIGS. **5**A, **5**B and **5**C, according to one inventive implementation. In essence, FIGS. **22**A and **22**B provide another perspective and summarize the various process flows and corresponding devices involved in the creation and provision of a live stream of digital content by a broadcaster to a media server, and the processing of the live stream by the media server. Although FIGS. **22**A and **22**B are directed primarily to the overall process flow for a

mobile broadcaster, the functionality and devices shown in these figures applies similarly to web-based broadcasters as well.

2) Viewer Processes

FIGS. **23**A and **23**B show a communication flow diagram illustrating process flow elements and the server and/or memory storage devices involved in the communication flow for a live stream RTMP media server or RTMP CDN viewer, according to one inventive implementation. A viewer who is a registered or anonymous user, but has received a StreamID for a particular broadcaster's live stream (e.g., via a push notification) to their viewer client device, may send a request to the web server(s) **700** (via the API) to receive a copy of the broadcaster's live stream. The web server(s) first checks the memory cache **460** for, or requests from the database **420**, various stream information corresponding to the StreamID provided by the requesting viewer. The web server(s) then perform(s) the viewer stream source selection algorithm discussed above in connection with FIG. **7** to provide an endpoint to the viewer client device for the appropriate media source from which to obtain a copy of the live stream. In the process shown in FIGS. **23**A and **23**B, the viewer stream source selection algorithm provides an endpoint (e.g., address or URL) to the viewer client device to establish a video communication channel with either a particular media server of the RTMP media server pool **320**, or a particular server of the RTMP CDN **340**.

The viewer client device also connects to the appropriate socket of the socket server(s) dedicated to the live stream to establish a chat/system event channel and thereby receive chat messages and system messages. If the live stream relates to an event, the viewer client device also connects to the appropriate socket of the socket server(s) dedicated to the event to establish an event information channel and thereby receive event messages containing various event information. The viewer using the viewer client device also may send chat messages to the web server API, which the web server directs to the appropriate socket of the socket server(s) dedicated to the live stream for broadcast to other viewers connected to the socket as well as the broadcaster. The web server also updates a replay log with the chat message from the viewer, so that the chat may be recreated if a recording of the broadcaster's live stream is replayed by a viewer at a later time (discussed further below).

FIGS. **24**A and **24**B show a communication flow diagram illustrating process flow elements and the server and/or memory storage devices involved in the communication flow for a live stream HLS viewer, according to one inventive implementation. The process shown in these figures is substantially similar to that outlined above in connection with FIGS. **23**A and **23**B; the primary difference is that, as a result of the web server(s) performing the viewer stream source selection algorithm (see FIG. **7**), the web server(s) return(s) to the viewer client device an endpoint (e.g., address or URL) to establish a video channel with the HLS server architecture **380** rather than a server of the RTMP media server pool **320** or the RTMP CDN **340**.

FIGS. **25**A, **25**B, and **25**C show a process flow illustrating a mobile client live stream replay method, according to one inventive implementation. For replay of a recording of a broadcaster's live stream, the servers and memory storage devices **1000** log all events that occur in connection with a live stream (e.g., chat messages and system event messages, as well as event message) and tie them to a timestamp. This allows synchronization of all events to the replay in the same order that the events occurred during the live stream, as if the viewer were not watching a recording of the live stream but actually watching a copy of the live stream in real time.

As shown in the figures, the viewer client device couples to the web server(s) via the API to request stream information and, if the stream recording is ready, loads the initial replay data from the API and then loads the media file of the recording. The viewer client device also connects to the chat/system event socket corresponding to the live stream (via a persistent authenticated connection), not to receive chat messages or system event messages (these messages are not present on replay), but rather so that the system knows of the viewer's presence and connection. Playback of the video is then started, and then the internal clock and the current video time clock are updated to provide for appropriate buffering of the video data. As the recording is played back, event data (e.g., chat messages, system messages, event information messages) is processed in one implementation according to FIGS. **26**A and **26**B, and user inputs are processed in one implementation according to FIG. **27**.

CONCLUSION

While various inventive implementations have been described and illustrated herein, those of ordinary skill in the art will readily envision a variety of other means and/or structures for performing the function and/or obtaining the results and/or one or more of the advantages described herein, and each of such variations and/or modifications is deemed to be within the scope of the inventive implementations described herein. More generally, those skilled in the art will readily appreciate that all parameters and configurations described herein are meant to be exemplary inventive features and that other equivalents to the specific inventive implementations described herein may be realized. It is, therefore, to be understood that the foregoing implementations are presented by way of example and that, within the scope of the appended claims and equivalents thereto, inventive implementations may be practiced otherwise than as specifically described and claimed. Inventive implementations of the present disclosure are directed to each individual feature, system, article, and/or method described herein. In addition, any combination of two or more such features, systems, articles, and/or methods, if such features, systems, articles, and/or methods are not mutually inconsistent, is included within the inventive scope of the present disclosure.

The above-described implementations can be implemented in multiple ways. For example, implementations may be implemented using hardware, software or a combination thereof. When implemented in software, the software code can be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers. Further, it should be appreciated that a computer may be embodied in any of a number of forms, such as a rack-mounted computer, a desktop computer, a laptop computer, or a tablet computer. Additionally, a computer may be embedded in a device not generally regarded as a computer but with suitable processing capabilities, including a Personal Digital Assistant (PDA), a smart phone or any other suitable portable or fixed electronic device.

Also, a computer may have one or more input and output devices. These devices can be used, among other things, to present a user interface. Examples of output devices that can be used to provide a user interface include printers or display screens for visual presentation of output and speakers or other sound generating devices for audible presentation of

output. Examples of input devices that can be used for a user interface include keyboards, and pointing devices, such as mice, touch pads, and digitizing tablets. As another example, a computer may receive input information through speech recognition or in other audible format. Such computers may be interconnected by one or more networks such as Internet. The various methods or processes outlined herein may be coded as software that is executable on one or more processors that employ any one of a variety of operating systems or platforms. Additionally, such software may be written using any of a number of suitable programming languages and/or programming or scripting tools, and also may be compiled as executable machine language code or intermediate code that is executed on a framework or virtual machine.

In this respect, various inventive concepts may be embodied as a computer readable memory or storage medium (or multiple computer readable storage media) (e.g., a computer memory, one or more floppy discs, compact discs, optical discs, magnetic tapes, flash memories, circuit configurations in Field Programmable Gate Arrays or other semiconductor devices, or other non-transitory medium or tangible computer storage medium) encoded with one or more programs that, when executed on one or more computers or other processors, perform methods that implement the various implementations of the invention discussed above. The computer readable medium or media can be transportable, such that the program or programs stored thereon can be loaded onto one or more different computers or other processors to implement various aspects of the present invention as discussed above.

Unless otherwise indicated, the terms "program" or "software" are used herein in a generic sense to refer to any type of computer code or set of computer-executable instructions that can be employed to program a computer or other processor to implement various aspects of implementations as discussed above. Additionally, it should be appreciated that according to one aspect, one or more computer programs that when executed perform methods of the present invention need not reside on a single computer or processor, but may be distributed in a modular fashion amongst a number of different computers or processors to implement various aspects of the present invention.

Computer-executable instructions may be in many forms, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various implementations.

Also, data structures may be stored in computer-readable media in any suitable form. For simplicity of illustration, data structures may be shown to have fields that are related through location in the data structure. Such relationships may likewise be achieved by assigning storage for the fields with locations in a computer-readable medium that convey relationship between the fields. However, any suitable mechanism may be used to establish a relationship between information in fields of a data structure, including through the use of pointers, tags or other mechanisms that establish relationship between data elements. In some implementations, a schema-minimal storage system may be implemented in a relational database environment using key-value storage versus defined data structures.

With the foregoing in mind, each of the client devices described herein, as well as various servers and other computing devices of the broadcast/viewing servers and memory storage devices shown for example in FIGS. **2** and **3**, may comprise one or more processors, one or more memory devices or systems communicatively coupled to the one or more processors (e.g., to store software code and other data), and one or more communication interfaces communicatively coupled to the one or more processors so as to implement the various specific and inventive functionality described herein.

Also, various inventive concepts may be embodied as one or more methods, of which an example has been provided. The acts performed as part of the method may be ordered in any suitable way. Accordingly, implementations may be constructed in which acts are performed in an order different than illustrated, which may include performing some acts simultaneously, even though shown as sequential acts in illustrative implementations.

All publications, patent applications, patents, and other references mentioned herein are incorporated by reference in their entirety.

All definitions, as defined and used herein, should be understood to control over dictionary definitions, definitions in documents incorporated by reference, and/or ordinary meanings of the defined terms.

The indefinite articles "a" and "an," as used herein in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean "at least one."

The phrase "and/or," as used herein in the specification and in the claims, should be understood to mean "either or both" of the elements so conjoined, i.e., elements that are conjunctively present in some cases and disjunctively present in other cases. Multiple elements listed with "and/or" should be construed in the same fashion, i.e., "one or more" of the elements so conjoined. Other elements may optionally be present other than the elements specifically identified by the "and/or" clause, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, a reference to "A and/or B", when used in conjunction with open-ended language such as "comprising" can refer, in one implementation, to A only (optionally including elements other than B); in another implementation, to B only (optionally including elements other than A); in yet another implementation, to both A and B (optionally including other elements); etc.

As used herein in the specification and in the claims, "or" should be understood to have the same meaning as "and/or" as defined above. For example, when separating items in a list, "or" or "and/or" shall be interpreted as being inclusive, i.e., the inclusion of at least one, but also including more than one, of a number or list of elements, and, optionally, additional unlisted items. Only terms clearly indicated to the contrary, such as "only one of" or "exactly one of," or, when used in the claims, "consisting of," will refer to the inclusion of exactly one element of a number or list of elements. In general, the term "or" as used herein shall only be interpreted as indicating exclusive alternatives (i.e. "one or the other but not both") when preceded by terms of exclusivity, such as "either," "one of," "only one of," or "exactly one of." "Consisting essentially of," when used in the claims, shall have its ordinary meaning as used in the field of patent law.

As used herein in the specification and in the claims, the phrase "at least one," in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in

the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list of elements to which the phrase "at least one" refers, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, "at least one of A and B" (or, equivalently, "at least one of A or B," or, equivalently "at least one of A and/or B") can refer, in one implementation, to at least one, optionally including more than one, A, with no B present (and optionally including elements other than B); in another implementation, to at least one, optionally including more than one, B, with no A present (and optionally including elements other than A); in yet another implementation, to at least one, optionally including more than one, A, and at least one, optionally including more than one, B (and optionally including other elements); etc.

In the claims, as well as in the specification above, all transitional phrases such as "comprising," "including," "carrying," "having," "containing," "involving," "holding," "composed of," and the like are to be understood to be open-ended, i.e., to mean including but not limited to. Only the transitional phrases "consisting of" and "consisting essentially of" shall be closed or semi-closed transitional phrases, respectively, as set forth in the United States Patent Office Manual of Patent Examining Procedures, Section 2111.03.

The invention claimed is:

1. A system, comprising:

at least one socket server; and

a web server configured to:

transmit first instructions to a first client device that includes at least one first display to cause the at least one first display of the first client device to render a first video relating to a first event and render first event information relating to the first event, wherein the first instructions transmitted by the webserver cause the first client device to:

receive, on a first communication channel, first digital content corresponding to the first video relating to the first event; and

receive, on a second communication channel different from the first communication channel, second digital content corresponding to the first event information from a first event socket of the at least one socket server; and

transmit second instructions to a second client device that includes at least one second display to cause the at least one second display of the second client device to render the first video relating to the first event and to render the first event information relating to the first event, wherein the second instructions transmitted by the webserver cause the second client device to:

receive, on a third communication channel, the first digital content corresponding to the first video relating to the first event; and

receive, on a fourth communication channel different from the third communication channel, the second digital content corresponding to the first event information,

wherein:

the second instructions transmitted by the webserver cause the second client device to receive the second digital content corresponding to the first event information via the fourth communication channel from the first event socket of the at least one socket server so that the second digital content received at the first client device and the second digital content received at the second client device is synchronized and client-by-client latency between the first client device and second client device to render the at least some of the first event information is thereby significantly reduced or mitigated.

2. The system of claim **1**, wherein:

the first instructions transmitted by the webserver cause the first client device to:

render, on the at least one first display of the first client device, the first video relating to the first event based on the first digital content received on the first communication channel; and

render, on the at least one first display of the first client device, at least some of the first event information based on the second digital content received on the second communication channel; and

the second instructions transmitted by the webserver cause the second client device to:

render, on the at least one second display of the second client device, the first video relating to the first event based on the first digital content received on the third communication channel; and

render, on the at least one second display of the second client device, at least some of the first event information based on the second digital content received on the fourth communication channel.

3. The system of claim **1**, further comprising:

at least one control server, coupled to the at least one socket server, to regularly update the second digital content provided to the first event socket of the at least one socket server so as to significantly reduce a first latency between the at least some of the first event information and the first video relating to the first event rendered on the at least one first display of the first client device and on the at least one second display of the second client device.

4. The system of claim **3**, wherein the at least one control server is configured to update the second digital content representing the first event information at least once a second to provide regular updates to the first event socket of the at least one socket server.

5. The system of claim **3**, wherein:

the first instructions transmitted by the webserver cause the first client device to render, on the at least one first display of the first client device, the first video relating to the first event with a first viewer latency on the order of three seconds or less; and

the second instructions transmitted by the webserver cause the second client device to render, on the at least one second display of the second client device, the first video relating to the first event with a second viewer latency on the order of three seconds or less.

6. The system of claim **1**, wherein:

the first instructions transmitted by the webserver cause the first client device to:

render, on the at least one first display of the first client device, at least some of the first event information as at least one first user-interactive feature overlaid on or adjacent to the first video relating to the first event; and

obtain access to first additional information relating to the first event and/or redirect to a first web site or first app when a first user selects at least a portion of the rendered at least one first user-interactive feature; and

the second instructions transmitted by the webserver cause the second client device to:

render, on the at least one second display of the second client device, at least some of the first event information as at least one second user-interactive feature; and

obtain access to the first additional information or second additional information relating to the first event and/or redirect to the first web site, the first app, a second web site or a second app related to the first event when a second user selects at least a portion of the rendered at least one second user-interactive feature.

7. The system of claim **6**, further comprising:

at least one control server, coupled to the at least one socket server, to regularly update the second digital content provided to the first event socket of the at least one socket server so as to significantly reduce a first latency between the at least some of the first event information and the first video relating to the first event rendered on the at least one first display of the first client device and on the at least one second display of the second client device.

8. The system of claim **6**, wherein:

the first event is a sporting or gaming event; and

the first event information is online gaming information.

9. The system of claim **1**, wherein:

the first instructions transmitted by the webserver cause the first client device to:

render, on a first portion of the at least one first display of the first client device, the first video relating to the first event based on the first copy of the audio/video feed received on the first communication channel; and

render, on a second portion of the at least one first display of the first client device, the at least some of the first event information based on the first digital content received on the second communication channel,

wherein the first portion is adjacent to the second portion or overlapping with the second portion.

10. A method, comprising:

A) transmitting first instructions to a first client device that includes at least one first display to cause the at least one first display of the first client device to render a first video relating to a first event and render first event information relating to the first event, wherein the first instructions transmitted in A) cause the first client device to:

receive, on a first communication channel, first digital content corresponding to the first video relating to the first event; and

receive, on a second communication channel different from the first communication channel, second digital content corresponding to the first event information from a first event socket of at least one socket server; and

B) transmitting second instructions to a second client device that includes at least one second display to cause the at least one second display of the second client device to render the first video relating to the first event and to render the first event information relating to the first event, wherein the second instructions transmitted in B) cause the second client device to:

receive, on a third communication channel, the first digital content corresponding to the first video relating to the first event; and

receive, on a fourth communication channel different from the third communication channel, the second digital content corresponding to the first event information,

wherein:

the second instructions transmitted in B) cause the second client device to receive the second digital content corresponding to the first event information via the fourth communication channel from the first event socket of the at least one socket server so that the second digital content received at the first client device and the second digital content received at the second client device is synchronized and client-by-client latency between the first client device and second client device to render the at least some of the first event information is thereby significantly reduced or mitigated.

11. The method of claim **10**, wherein:

the first instructions transmitted in A) cause the first client device to:

render, on the at least one first display of the first client device, the first video relating to the first event based on the first digital content received on the first communication channel; and

render, on the at least one first display of the first client device, at least some of the first event information based on the second digital content received on the second communication channel; and

the second instructions transmitted in B) cause the second client device to:

render, on the at least one second display of the second client device, the first video relating to the first event based on the first digital content received on the third communication channel; and

render, on the at least one second display of the second client device, at least some of the first event information based on the second digital content received on the fourth communication channel.

12. The method of claim **10**, further comprising:

regularly updating the second digital content provided to the first event socket of the at least one socket server so as to significantly reduce a first latency between the at least some of the first event information and the first video relating to the first event rendered on the at least one first display of the first client device and on the at least one second display of the second client device.

13. The method of claim **12**, wherein regularly updating the second digital content provided to the first event socket of the at least one socket server comprises:

updating the second digital content representing the first event information at least once a second to provide regular updates to the first event socket of the at least one socket server.

14. The method of claim **12**, wherein:

the first instructions transmitted in A) cause the first client device to render, on the at least one first display of the first client device, the first video relating to the first event with a first viewer latency on the order of ten seconds or less; and

the second instructions transmitted in B) cause the second client device to render, on the at least one second display of the second client device, the first video relating to the first event with a second viewer latency on the order of ten seconds or less.

15. The method of claim **12**, wherein:

the first instructions transmitted in A) cause the first client device to render, on the at least one first display of the

first client device, the first video relating to the first event with a first viewer latency on the order of three seconds or less; and

the second instructions transmitted in B) cause the second client device to render, on the at least one second display of the second client device, the first video relating to the first event with a second viewer latency on the order of three seconds or less.

16. The method of claim **12**, wherein:

the first instructions transmitted in A) cause the first client device to render, on the at least one first display of the first client device, the first video relating to the first event with a first viewer latency on the order of two seconds or less; and

the second instructions transmitted in B) cause the second client device to render, on the at least one second display of the second client device, the first video relating to the first event with a second viewer latency on the order of two seconds or less.

17. The method of claim **12**, wherein:

the first instructions transmitted in A) cause the first client device to:

render, on the at least one first display of the first client device, at least some of the first event information as at least one first user-interactive feature overlaid on or adjacent to the first video relating to the first event; and

obtain access to first additional information relating to the first event and/or redirect to a first web site or first app when a first user selects at least a portion of the rendered at least one first user-interactive feature; and

the second instructions transmitted in B) cause the second client device to:

render, on the at least one second display of the second client device, at least some of the first event information as at least one second user-interactive feature; and

obtain access to the first additional information or second additional information relating to the first event and/or redirect to the first web site, the first app, a second web site or a second app related to the first event when a second user selects at least a portion of the rendered at least one second user-interactive feature.

18. The method of claim **12**, wherein:

the first instructions transmitted in A) cause the first client device to:

render, on a first portion of the at least one first display of the first client device, the first video relating to the first event based on the first copy of the audio/video feed received on the first communication channel; and

render, on a second portion of the at least one first display of the first client device, the at least some of the first event information based on the first digital content received on the second communication channel.

19. The method of claim **18**, wherein the first instructions transmitted in A) cause the first portion of the at least one first display of the first client device to be adjacent to the second portion of the at least one first display of the first client device.

20. The method of claim **18**, wherein the first instructions transmitted in A) cause the first portion and the second portion of the at least one first display of the first client device to be at least partially overlapping.

21. The method of claim **18**, wherein the first instructions transmitted in A) cause the second portion of the at least one first display of the first client device to include only at least one text element and/or at least one graphics element.

22. The method of claim **18**, wherein the first instructions transmitted in A) cause the second portion of the at least one first display of the first client device to include at least one interactive element.

23. The method of claim **22**, wherein the first instructions transmitted in A) cause the first client device, when a user of the first client device interacts with the at least one interactive element, to:

launch further graphics or animations;

receive additional information about the first event; and/or navigate to an Internet location.

24. The method of claim **22**, wherein the first instructions transmitted in C) cause the first client device, when a user of the first client device interacts with the at least one interactive element, to navigate to an Internet location.

25. The method of claim **22**, wherein:

the first event is a sporting or gaming event; and

the first event information is online gaming information.

26. A system, comprising:

at least one socket server;

a web server configured to:

transmit first instructions to a first client device that includes at least one first display to cause the at least one first display of the first client device to render a first video relating to a first event and render first event information relating to the first event, wherein the first instructions transmitted by the webserver cause the first client device to:

receive, on a first communication channel, first digital content corresponding to the first video relating to the first event; and

receive, on a second communication channel different from the first communication channel, second digital content corresponding to the first event information from a first event socket of the at least one socket server; and

transmit second instructions to a second client device that includes at least one second display to cause the at least one second display of the second client device to render the first video relating to the first event and to render the first event information relating to the first event, wherein the second instructions transmitted by the webserver cause the second client device to:

receive, on a third communication channel, the first digital content corresponding to the first video relating to the first event; and

receive, on a fourth communication channel different from the third communication channel, the second digital content corresponding to the first event information from the first event socket of the at least one socket server; and

at least one control server, coupled to the at least one socket server, to regularly update the second digital content provided to the first event socket of the at least one socket server so as to significantly reduce a first latency between the at least some of the first event information and the first video relating to the first event.

27. The system of claim **26**, wherein the at least one control server is configured to update the second digital content representing the first event information at least once a second to provide regular updates to the first event socket of the at least one socket server.

28. The system of claim 26, wherein:

the first instructions transmitted by the webserver cause the first client device to:

render, on the at least one first display of the first client device, at least some of the first event information as at least one first user-interactive feature overlaid on or adjacent to the first video relating to the first event; and

obtain access to first additional information relating to the first event and/or redirect to a first web site or first app when a first user selects at least a portion of the rendered at least one first user-interactive feature; and

the second instructions transmitted by the webserver cause the second client device to:

render, on the at least one second display of the second client device, at least some of the first event information as at least one second user-interactive feature; and

obtain access to the first additional information or second additional information relating to the first event and/or redirect to the first web site, the first app, a second web site or a second app related to the first event when a second user selects at least a portion of the rendered at least one second user-interactive feature.

29. The system of claim 28, wherein:

the first event is a sporting or gaming event; and

the first event information is online gaming information.

30. The system of claim 29, wherein:

the first instructions transmitted by the webserver cause the first client device to render, on the at least one first display of the first client device, the first video relating to the first event with a first viewer latency on the order of two seconds or less; and

the second instructions transmitted by the webserver cause the second client device to render, on the at least one second display of the second client device, the first video relating to the first event with a second viewer latency on the order of two seconds or less.

* * * * *