

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC., and
AMAZON WEB SERVICES, INC.,

Petitioners,

v.

DIVX, LLC,

Patent Owner.

Case: IPR2025-01222

U.S. Patent No. 10,412,141

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,412,141**

TABLE OF CONTENTS

	Page
I. Requirements for IPR Under 37 C.F.R. § 42.104	1
A. Standing (37 C.F.R. § 42.104(a)).....	1
B. Identification of Challenge (37 C.F.R. § 42.104(b)).....	1
C. Claim Construction (37 C.F.R. § 42.104(b)(3)).....	2
II. Overview of the 141 Patent	2
A. Technical Background.....	2
1. Progressive Playback Systems.....	2
a. Architecture of Progressive Playback Systems	3
b. Media Files	4
c. Trick Play and Indexing	5
d. Encryption and Digital Rights Management.....	6
B. Patent Description	6
C. Priority Date	7
D. Level of Ordinary Skill in the Art.....	7
III. Prior Art Overview	8
A. Hagai.....	8
B. Li.....	12
C. Park.....	14
D. Schmitz.....	14

TABLE OF CONTENTS **(Continued)**

	Page
IV. GROUND 1: Claims 1, 3-8, 10, 20, 22-27, and 29 are obvious over Hagai, Li, and General Knowledge of a POSITA	15
A. A POSITA would have combined Hagai and Li.....	15
B. Claim 1	18
1. [1.pre] “A playback device, comprising:”	18
2. [1.a] “a processor; and”	20
3. [1.b] “a non-volatile storage containing an application for causing the processor to perform the steps of:”	21
4. [1.b(i)] “establishing at least one connection for communicating with a remote server system;”	21
5. [1.b(ii)] “obtaining information from a remote server system describing at least one video track, multiple audio tracks, and multiple subtitle tracks;”	23
6. [1.b(iii)] “selecting a video track from the at least one video track;”	29
7. [1.b(iv)] “requesting a header describing the selected video track;”	29
8. [1.b(v)] “selecting an audio track from the multiple audio tracks;”	31
9. [1.b(vi)] “obtaining index information indicating the locations of audio and video data within the selected audio and video tracks;”	32
10. [1.b(vii)] “determining byte ranges to request from the selected audio and video tracks using the index information;”	35

TABLE OF CONTENTS (Continued)

		Page
11.	[1.b(viii)] “requesting byte ranges from the selected video track and the selected audio track from the remote server system;”	36
12.	[1.b(ix)] “buffering received bytes of information comprising audio and video data;”	37
13.	[1.b(x)] “checking that sufficient data is buffered to commence playback and playing back the buffered audio and video data;”	38
14.	[1.b(xi)] “responding to a received seek instruction by;”	38
15.	[1.b(xi)(1)] “pausing playback;”	39
16.	[1.b(xi)(2)] “determining byte ranges to request from the selected audio and video tracks based upon a new playback location using the index information;”	40
17.	[1.b(xi)(3)] “requesting byte ranges required to play the selected audio and video tracks from the new playback location from the remote server;”	40
18.	[1.b(xi)(4)] “buffering received bytes of information comprising audio and video data pending resumption of playback; and”	40
19.	[1.b(xi)(5)] “checking that sufficient data is buffered to commence playback and playing back the buffered audio and video data.”	40
C.	Claim 3: “The playback device of claim 1, wherein the step of requesting a header describing the selected video track comprises the step of requesting a DRM header.”	40

TABLE OF CONTENTS **(Continued)**

	Page
D. Claim 4: “The playback device of claim 3, wherein the application is further capable of causing the processor to perform the step of decrypting the DRM header.”	43
E. Claim 5	45
1. [5.pre], [5.a] “The playback device of claim 3, wherein: the DRM header identifies at least one key; and”	45
2. [5.b] “the application is further capable of causing the processor to perform the steps of decrypting encrypted frames of video using the at least one key prior to decoding.”	46
F. Claim 6	48
1. [6.pre], [6.a] “The playback device of claim 5, wherein: encrypted frames of video are only partially encrypted;”	48
2. [6.b] “the received bytes of information contain references to portions of video frames that are encrypted; and”	50
3. [6.c] the step of decrypting encrypted frames of video using the at least one key prior to decoding further comprises decrypting encrypted portions of frames of video using the references to the portions of the video frames that are encrypted.”	50
G. Claim 7: “The playback device of claim 1, wherein playing back the buffered video data from the requested byte ranges provides fast forward trick play.”	52
H. Claim 8: “The playback device of claim 1, wherein playing back the buffered video data from the requested byte ranges provides fast rewind trick play.”	53

TABLE OF CONTENTS
(Continued)

	Page
I. Claim 10	54
1. [10.pre], [10.a]: “The playback device of claim 1, wherein the application is further capable of causing the processor to perform the steps of: selecting a subtitle track from the multiple subtitle tracks;”	54
2. [10.b]: “requesting data from the selected subtitle track from the remote server system, where the data contains subtitle text; and”	55
3. [10.c]: “superimposing the subtitle text over video during playback.”	56
J. Claim 20:	57
1. [20.pre]	57
2. [20.a]–[20.k] and [20.k(i)]	57
3. [20.k(ii)]–[20.k(v)]	58
K. Claim 22	58
L. Claim 23	58
M. Claim 24	59
N. Claim 25	59
O. Claim 26	59
P. Claim 27	60
Q. Claim 29	60
V. GROUND 2: Claims 9, 12-14, 16-18 and 28 are obvious over Hagai, Li, Park, and General Knowledge of a POSITA.....	60
A. Motivation to Combine	60

TABLE OF CONTENTS
(Continued)

	Page
B. Claims 9 and 28.....	62
1. Claim 9: “The playback device of claim 1, wherein the application is further capable of causing the processor to perform the step of discarding audio and video data contained within the buffer in response to a received seek instruction.”	62
C. Claim 12	64
1. [12.pre].....	64
2. [12.a]	64
3. [12.b]	64
4. [12.b(i)]	64
5. [12.b(ii)]	64
6. [12.b(iii)]	64
7. [12.b(iv)]	65
8. [12.b(v)]	65
9. [12.b(vi)]	65
10. [12.b(vii)]	65
11. [12.b(viii)]	65
12. [12.b(ix)]	65
13. [12.b(x)]	66
14. [12.b(xi)]	66
15. [12.b(xii)]	66

TABLE OF CONTENTS (Continued)

	Page
16. [12.b(xiii)]	66
17. [12.b(xiv)]	67
18. [12.b(xv)]	67
19. [12.b(xv)(1)]	67
20. [12.b(xv)(2)]	67
21. [12.b(xv)(3)]–[12.b(xv)(8)]	67
VI. GROUND 3: Claims 2, 11, 21, and 30 are obvious over Hagai, Li, Schmitz, and General Knowledge of a POSITA	69
A. Motivation to Combine	69
B. Claims 11 and 30	72
1. Claim 11: “The playback device of claim 1, wherein the application is further capable of causing the processor to perform the step of flushing previous byte range requests in response to the received seek instruction.”	72
C. Claims 2 and 21	74
1. Claim 2: “The playback device of claim 1, wherein the application is further capable of causing the processor to asynchronously request byte ranges from the selected video track and the selected audio track from the remote server.”	74
VII. GROUND 4: Claims 15 and 19 are obvious over Hagai, Li, Park, Schmitz, and General Knowledge of a POSITA	76
A. Motivation to Combine	76
B. Claim 15	77

TABLE OF CONTENTS
(Continued)

	Page
C. Claim 19	77
VIII. CONCLUSION.....	78
IX. MANDATORY NOTICES AND FEES	78
A. Real Party-in-Interest	78
B. Related Matters.....	78
C. Lead and Backup Counsel.....	78
D. Service Information.....	79
E. Fees.....	79
APPENDIX: LIST OF CHALLENGED CLAIMS	82

TABLE OF AUTHORITIES

	Page(s)
STATUTES AND RULES	
35 U.S.C. § 102	8, 12, 13, 14
35 U.S.C. § 103	1, 2
35 U.S.C. § 311	1
OTHER AUTHORITIES	
37 C.F.R. § 42.100	1
37 C.F.R. § 42.104	1, 2, 3, 4

EXHIBIT LIST (37 C.F.R. § 42.63(e))

Ex. No.	Description
1001	U.S. Patent No. 10,412,141 (“the 141 patent”)
1002	Excerpts of Prosecution File History for the 141 Patent
1003	Declaration of Hari Kalva, Ph.D.
1004	U.S. Patent No. 7,051,110 (Hagai)
1005	U.S. Patent No. 8,731,369 (Li)
1006	U.S. Patent Application Publication No. 2005/0198681 (Park)
1007	U.S. Patent No. 5,544,318 (Schmitz)
1008	Poole, C. & Bradley, J., “Developer’s Digital Media Reference,” Focal Press (2003) (Poole)
1009	U.S. Patent Application Publication No. 2005/0102371 (Aksu)
1010	U.S. Patent No. 7,962,942 (Craner)
1011	U.S. Patent No. 5,874,986 (Gibbon)
1012	Apple Computer, Inc., “QuickTime 7 User’s Guide” (2005) (QuickTime 7 User’s Guide)
1013	Farber, N. et al., “Adaptive progressive download based on the MPEG-4 file format,” J. Zhejiang Univ. 2006 7 (Suppl. I) 106-111 (Farber)
1014	U.S. Patent Application Publication No. 2006/0037057 (Xu)
1015	U.S. Patent Application Publication No. 2005/0108320 (Lord)
1016	Sun, H. et al., “Digital Video Transcoding for Transmission and Storage,” CRC Press (2005) (Sun)
1017	U.S. Patent No. 6,754,715 (Cannon)

Ex. No.	Description
1018	U.S. Patent No. 6,965,646 (Firestone)
1019	U.S. Patent No. 7,826,712 (Minnick)
1020	U.S. Patent No. 5,566,208 (Balakrishnan)
1021	U.S. Patent No. 7,073,063 (Peinado)
1022	U.S. Patent No. 7,376,233 (Candelore)
1023	U.S. Patent No. 5,477,263 (O’Callaghan)
1024	U.S. Patent No. 5,878,135 (Blatter)
1025	U.S. Patent No. 7,647,614 (Krikorian)
1026	U.S. Patent Application Publication No. 2001/0030710 (Werner)
1027	U.S. Patent Application Publication No. 2005/0078947 (Chung)
1028	Mahemoff, M., “Ajax Design Patterns: Creating Web 2.0 Sites with Programming and Usability Patterns,” O’Reilly (2006) (Mahemoff)
1029	International Telecommunication Union, “ITU-T Rec. H.262 – Information technology – Generic coding of moving pictures and associated audio information: Video,” (Feb. 2000)
1030	Fielding, R. et al., “RFC 2616 – Hypertext Transfer Protocol – HTTP/1.1,” Network Working Group (June, 1999) (RFC2616)
1031	U.S. Patent No. 7,739,351 (Shkvarchuk)
1032	U.S. Patent No. 8,238,725 (Demas)
1033	European Patent Application No. EP 1 158 799 (Rittner)
1034	European Patent Application No. EP 1 343 321 (England)
1035	Curriculum Vitae of Hari Kalva, Ph.D.

Pursuant to 35 U.S.C. § 311 and 37 C.F.R. § 42.100, Amazon.com, Inc. and Amazon Web Services, Inc. (“Petitioner”) requests *Inter Partes* Review (“IPR”) of claims 1-30 (the “Challenged Claims”) of U.S. Patent No. 10,412,141 (the “141 patent”), assigned to DivX, LLC. The Challenged Claims are obvious over the prior art. Accordingly, the Patent Trial and Appeal Board should institute review of the Challenged Claims.

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

A. Standing (37 C.F.R. § 42.104(a))

Petitioner certifies that the 141 patent is available for IPR and that Petitioner is not barred or estopped from requesting IPR of the Challenged Claims.

B. Identification of Challenge (37 C.F.R. § 42.104(b))

In view of the prior art and evidence presented, the Challenged Claims are unpatentable and should be cancelled. 37 C.F.R. § 42.104(b)(1). Based on the prior art identified below, IPR of the Challenged Claims should be instituted. 37 C.F.R. § 42.104(b)(2).

Proposed Grounds of Unpatentability	
Ground 1	Claims 1, 3-8, 10, 20, 22-27, and 29 are obvious under pre-AIA § 103 over Hagai, Li, and General Knowledge of a POSITA
Ground 2	Claims 9, 12-14, 16-18, and 28 are obvious under pre-AIA § 103 over Hagai, Li, Park, and General Knowledge of a POSITA

Ground 3	Claims 2, 11, 21, and 30 are obvious under pre-AIA § 103 over Hagai, Li, Schmitz, and General Knowledge of a POSITA
Ground 4	Claims 15 and 19 are obvious under pre-AIA § 103 over Hagai, Li, Park, Schmitz and General Knowledge of a POSITA

Sections IV-VII identify where each element of the Challenged Claims is found in the prior art. 37 C.F.R. § 42.104(b)(4). The exhibit numbers of the evidence relied on to support the challenges are provided above, and the relevance of the evidence to the challenges raised are provided in Sections II-III. 37 C.F.R. § 42.104(b)(5). Exhibits **1001-1035** are attached.

C. Claim Construction (37 C.F.R. § 42.104(b)(3))

For the purposes of this Petition, all claims are interpreted with their ordinary and customary meaning as understood from the perspective of a person of ordinary skill in the art (“POSITA”).¹

II. OVERVIEW OF THE 141 PATENT

A. Technical Background

1. Progressive Playback Systems

The 141 patent is directed to a progressive video playback system with support for play operations (e.g., fast-forward, rewind, and seek). EX1001, 2:23-39; EX1003, ¶30. Progressive playback was a well-known technique for playing video

¹ Petitioner reserves the right to offer different claim constructions in other forums.

content while still being downloaded, rather than requiring the entire file to be downloaded first. EX1008, 6; EX1003, ¶30. In progressive playback, a server delivers video over a network to a playback device in small, sequential chunks (i.e., portions of the multimedia file). EX1003, ¶30. The playback device buffers each chunk as it arrives and plays it back once sufficient data is buffered. *Id.*

a. Architecture of Progressive Playback Systems

Progressive download and trick play were well-known in the art before 2007. EX1003, ¶31. Both required at least a remote server storing multimedia data and a playback device which requests and receives data from the server. EX1009, [0015]; *infra* § II.A.1.b; EX1003, ¶37. Known playback devices at the time included personal computers, mobile phones, and other devices that could communicate with a server over a network and playback data received. EX1010, 3:20-37; EX1004, 1:17-20; EX1001, 6:10-20; EX1003, ¶31. All these devices included at least a processor for executing code, a memory buffer for storing received stream data, and mechanisms for implementing trick play. EX1011, 5:51-56; EX1012, 8-9; EX1005, 2:53-59; EX1003, ¶31.

To access content, a playback device must establish a connection with a server via, for example, HTTP (Hypertext Transfer Protocol). EX1001, 6:38-46; EX1004, 1:46-52; EX1009, [0030]; EX1003, ¶32. HTTP enables streaming systems to leverage existing internet infrastructure. EX1013, 4; EX1009, [0030]; EX1014,

[0014]. Once the playback device establishes a server connection, it may request information describing the content stored on the server, and based on the received information, request specific content for playback. EX1009, [0008]; EX1014, [0016]; EX1004, 6:59-62, 18:30-34; EX1015, [0033]; EX1003, ¶32.

b. Media Files

Media may be segmented for storage on the server. EX1003, ¶34. Segmentation divides a file into smaller pieces (or “chunks”), allowing it to be sent in manageable portions. EX1013, 4. For example, files using RIFF (Resource Interchange File Format), like Microsoft’s AVI format, stored pieces of audio, video, and other information in “chunks” that could be retrieved and processed individually. EX1005, 1:54-65. Segmenting media content allows requests for byte ranges corresponding to specific chunks without needing to download the entire file. EX1013, 4.

For efficient transmission over the internet, multimedia files are generally compressed through encoding. EX1003, ¶35. Uncompressed files are often too large for rapid delivery. EX1004, 1:46-52; EX1016, 64-65. To balance compression efficiency and playback fidelity, video data was often encoded in formats using I-, P-, and B-frames. EX1016, 64-65; EX1017, 10:66-11:6. Only I-frames, or “key frames,” could be decoded independently, making them valid entry points into the stream. *Id.*; EX1003, ¶36. As discussed below, index information requested by the

playback device identifies the locations of these key frames and enables almost instantaneous decoding and playback from arbitrary locations. EX1009, [0008]; EX1014, [0018].

c. Trick Play and Indexing

Trick play operations are user-controlled playback features, like fast-forward, rewind, pause, and skip, that mimic the interactivity of physical media like DVDs. EX1014, [0002]. Trick play functionality was well-known prior to 2007. EX1003, ¶40. For instance, Apple's QuickTime 7 player supported "floating DVD-like controls for easy access to functions like pause, play, fast-forward, and rewind." EX1012, 8.

To implement trick play operations, playback devices request index information from the server to identify and access relevant portions of a multimedia file. EX1014, [0016]; EX1009, [0010]; EX1018, 10:50-53, 13:35-37, 14:41-45; EX1003, ¶41. The index information may include details such as byte offsets identifying positions where playback can resume after a trick play command, like the locations of key frames. EX1014, [0016].

As explained above, the playback device stores received data in a buffer, from which it may be played back. EX1004, 10:4-10; EX1010, 7:26-30; EX1019, 4:26-29. The playback device manages the data levels of the buffer so that enough

buffered data is available to cushion against potential transmission delays and maintain a smooth streaming experience. EX1020, 1:46-52; EX1003, ¶42.

d. Encryption and Digital Rights Management

Many playback systems use digital rights management (DRM) to protect digital content by restricting access to authorized users. EX1021, 1:36-58. DRM commonly includes encryption to convert media content into secure, unreadable data that can only be accessed using a decryption key. EX1021, 2:47-51; EX1003, ¶44. For example, video data may be fully encrypted, or it may be partially encrypted such that only portions of certain frames, such as portions of key frames, are encrypted. EX1022, 13:21-30; EX1003, ¶45. Encrypted data cannot be decoded for playback without first decrypting it. EX1023, 3:62-65; EX. 1003, ¶48.

B. Patent Description

The 141 patent is directed to a progressive playback system that uses a client application to support “trick play” functions. According to the 141 patent, prior progressive playback systems could play back content as it was downloaded from a remote server. EX1001, 1:32-35; EX1003, ¶49. However, these prior systems typically only supported short video clips because they downloaded files linearly. EX1001, 1:40-45, 2:1-3. Because they could not request data out of order, they did not support “random seeking, trick-play and playback of remotely stored longer content such as feature length movies.” EX1001 at 1:51-55, 2:10-19; EX1003, ¶50.

The 141 patent purportedly attempts to overcome these limitations by using index information referencing data within the media file to allow the playback system to request specific byte ranges in the file from a remote server. EX1001, 2:48-52; EX1002, Response to OA dated February 20, 2019, pg. 16. As noted above, however, it was well-known to a POSITA that playback systems could support trick play functionality on partially downloaded media files using index information and requests for specific byte ranges of data. *Supra* § II.A.1.c.

C. Priority Date

The 141 patent claims priority to prior applications, the earliest of which was filed on January 5, 2007. EX1001, Cover. For purposes of this proceeding only, Petitioner does not dispute that the priority date is January 5, 2007. EX1003, ¶52.

D. Level of Ordinary Skill in the Art

A POSITA would have a bachelor's or master's degree in electrical engineering, computer engineering, or a related field, with specialized training or coursework in digital video transmission and playback and related technologies. The individual would have at least four years of practical experience in digital video transmission and playback, including software and/or hardware architectures used to perform digital video transmission and playback. A person could also have qualified as a POSITA with more formal education and less technical experience, or vice versa. EX1003, ¶53.

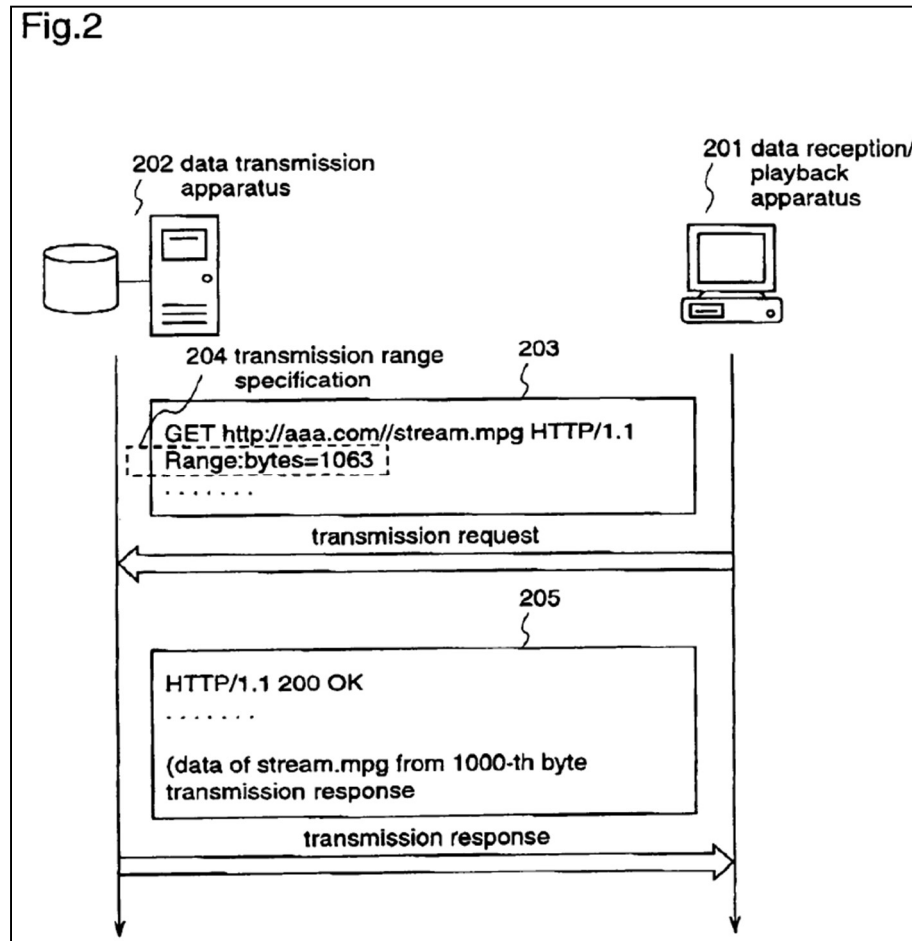
III. PRIOR ART OVERVIEW²

A. Hagai

U.S. Patent No. 7,051,110 to Makoto Hagai et al. was filed on December 20, 2000, published on August 9, 2001, and issued on May 23, 2006 (“Hagai,” EX1004). EX1004, Cover. Hagai is prior art to the 141 patent under pre-AIA §§ 102(a) and 102(b). Hagai was not disclosed to the Patent Office during prosecution of the 141 patent. EX1001, References Cited; EX1003, ¶54.

Hagai describes a client-server streaming system that enables seeking and other trick play operations using a playback start position table. EX1004, 8:1-10. In Hagai, a data reception/playback apparatus (“playback device”) communicates with a data transmission apparatus (“server”) over the internet using HTTP. EX1004, 1:25-28, 3:62-4:5, Fig. 2; EX1003, ¶55.

² Pre-AIA 35 U.S.C. § 102 applies to the 141 patent.



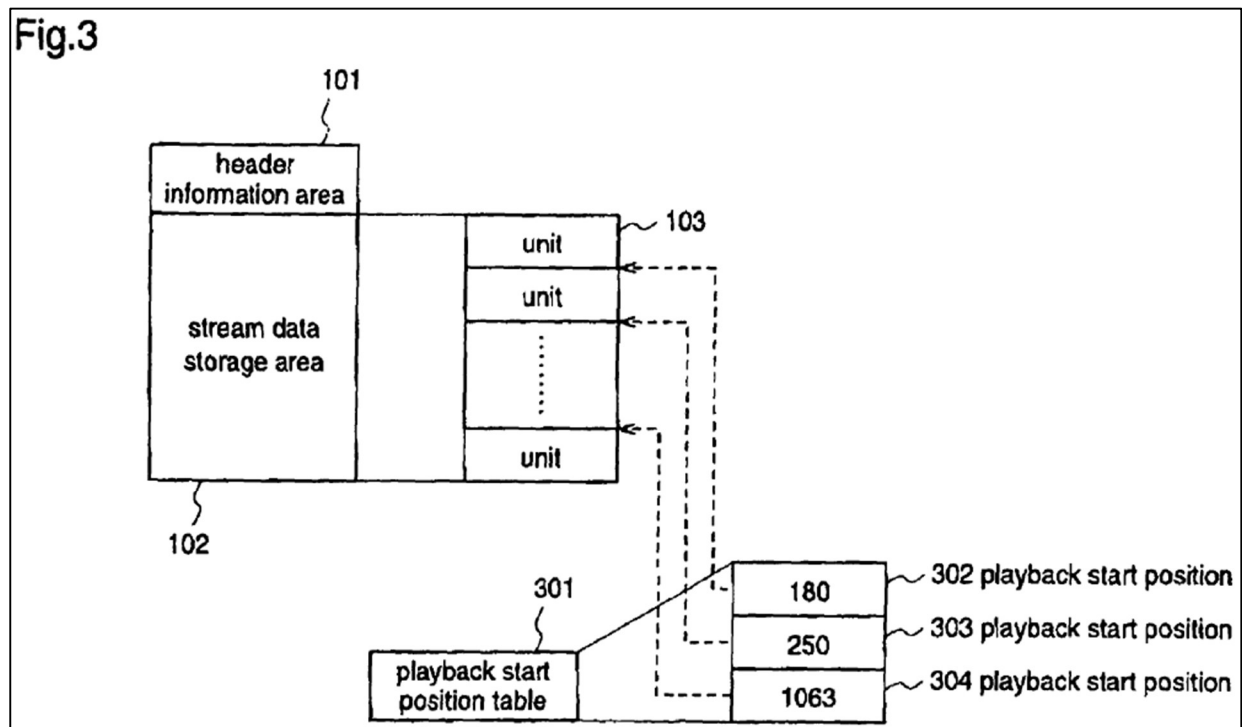
The playback device uses HTTP to send a request identifying the media file or portions of the media file. EX1004, 9:17-23, Fig. 2; EX1003, ¶56. In response, the server retrieves and sends a response with the requested data to be played back on the playback device. EX1004, 10:4-10; EX1003, ¶56.

The playback device sends other requests to the server for data related to the media files stored on the server, including a playback start position table. EX1004, 9:17-41; EX1003, ¶57. The playback start position table informs subsequent media requests by identifying byte locations corresponding to desired portions of content. EX1004, 9:47-54; EX1003, ¶57.

The playback device also includes a buffer that stores received data for playback, and a buffer monitor that monitors the quantity of buffered data. EX1004, 6:7-14, 10:4-10; EX1003, ¶58. If the buffer monitor determines that the buffer does not contain a sufficient level of data, playback will not continue. EX1004, 6:7-27, 16:48-51.

As noted above, the server stores media files and metadata related to the audio and video tracks of the stored media files. EX1004, 6:49-65, Fig. 15; EX1003, ¶59. These media files also include a header information area 101 and a stream data storage area 102. EX1004, 1:53-58, 2:56-58, 4:13-16, 8:11-18, Fig. 1; EX1003, ¶60.

According to one embodiment, the playback start position table 301 is also stored on the server, separately from the stream data, and identifies the number of bytes from the head position of the stream data storage area 102 to different playback start positions. EX1004, 8:41-59, Fig. 3; EX1003, ¶61.



Hagai's server also stores a playback-speed/file mapping table that identifies the playback speeds of files stored on the server. EX1004, 18:34-44, Fig. 16; EX1003, ¶62.

Fig.16

1601 playback speed		1602 file name	
1.0		stream.mpg	
2.0		stream_s2.mpg	
3.0		stream_s3.mpg	

Hagai discloses five embodiments generally directed to transmitting, receiving, and playing audio and video data through a network. EX1004, 1:9-12. Hagai's first and second embodiments relate to a foundational method of multimedia playback from arbitrary points in the stream (i.e., valid entry points in the stream selected by the user), and the other three embodiments describe additional operations supplementing this basic functionality. EX1004, 8:1-4, EX1003, ¶64. The third embodiment adds a halt/restart data reception operation. EX1004, 15:23-32. The fourth embodiment adds a buffer monitor function that prevents degradation in stream quality. EX1004, 16:24-30. The fifth embodiment adds a playback-speed/file mapping table that the playback device uses for video track selection. EX1004, 17:55-59.

B. Li

U.S. Patent No. 8,731,369 to Adam H. Li et al. was filed on December 17, 2004, published on September 22, 2005, and issued on May 20, 2014 ("Li," EX1005). Li is prior art to the 141 patent under pre-AIA §§ 102(a) and 102(b). The 141 patent cites Li as prior art, and the examiner cites Li (referred to as "Zoest") as a primary reference in the Office Action dated November 19, 2018. EX1001, Cover; EX1002, Office Action and Applicant's Response; EX1003, ¶65. To overcome the Office Action, the applicant amended the claims to distinguish over the other prior art in the examiner's combination and argued there was no motivation to combine.

Id. Here, Petitioner relies on a new combination, with Hagai as the primary reference.

Li discloses a system for “encoding, transmission and decoding of multimedia files that can include” multiple tracks. EX1005, 1:19-22. Li’s playback device includes a processor configured to extract metadata, locate video and audio chunks, decode video, identify corresponding subtitle and audio tracks, and superimpose subtitle text. EX1005, 2:26-67, Claims 1-8; EX1003, ¶66.

In Li, the multimedia file is structured as a RIFF file composed of data chunks. EX1005, 12:46-63. These chunks include DMNU chunks, DRM chunks, and DRM headers. EX1005, 23:3-12, 52:41-55; EX1003, ¶67.

A DMNU chunk contains AudioTrack and SubtitleTrack chunks that identify available audio and subtitle tracks. EX1005, 23:67-24:9. These chunks accompany portions of video identified in corresponding MediaTrack chunks. EX1005, 24:2-4, 25:27-30; EX1003, ¶68. The DMNU chunk also includes AudioSelectAction and SubtitleSelectAction chunks, which enable selection of user-desired tracks. EX1005, 24:19-31. Li’s multimedia files also include DRM chunks, which contain references to encrypted portions of video, and DRM headers, which contain the corresponding decryption keys. EX1005, 27:62-28:2; EX1003, ¶69.

C. Park

U.S. Patent Application Publication No. 2005/0198681 to Daniel John Park, was filed on March 8, 2004 and published on September 8, 2005 (“Park,” EX1006). EX1006, Cover; EX1003, ¶70. Park is prior art to the 141 patent under pre-AIA §§ 102(a) and 102(b). Park was not disclosed to the Patent Office during prosecution of the 141 patent. EX1001, References Cited.

Park discloses techniques for “managing a playout buffer to minimize delays” in response to user fast-forward, reverse, and pause commands. EX1006, [0001], [0012]. For example, Park discloses managing the buffer by flushing stored audio and video data in response to a user command, such as a fast-forward or reverse, or a change in AV program. EX1006, [0012], [0026]; Claim 7. Park explains that flushing the playout buffer minimizes playout delay by discarding outdated audio and video data. EX1006, [0023], [0027]; EX1003, ¶71.

D. Schmitz

U.S. Patent No. 5,544,318 to Patrick Schmitz et al., was filed on April 16, 1993, and issued on August 6, 1996 (“Schmitz,” EX1007). EX1007, Cover. Schmitz is prior art to the 141 patent under pre-AIA §§ 102(a) and 102(b). Schmitz was not disclosed to the Patent Office during prosecution of the 141 patent. EX1001, References Cited; EX1003, ¶72.

Schmitz is directed to navigation through multimedia content. EX1007, 6:18-22. Schmitz proposes a method of using asynchronous requests to enable arbitrary navigation through content rather than imposing linear constraints. EX1007, 5:63-65; EX1003, ¶73. To reduce unnecessary bandwidth use, Schmitz discloses flushing or cancelling any pending requests made obsolete when a user navigates to a new playback location. EX1007, 1:10-14, 6:62-65, Abstract; EX1003, ¶73.

While the technique of Schmitz has special application to non-linear media (like encyclopedias), its techniques also expressly apply to time-linear media, like video, that is ordinarily played back sequentially. EX1007, 1:67-2:7, 6:1-10; EX1003, ¶74.

IV. GROUND 1: Claims 1, 3-8, 10, 20, 22-27, and 29³ are obvious over Hagai, Li, and General Knowledge of a POSITA

A. A POSITA would have combined Hagai and Li

A POSITA would have been motivated to combine Hagai and Li. Both relate to transmission and processing of multimedia content for playback. EX1003, ¶76.

Hagai relates to “methods and apparatuses for transmitting, receiving, and playing audio data or video data (moving pictures, still pictures, graphics, etc.) through a network.” EX1004, 1:9-12. A POSITA would have understood that audio and video data are both components of multimedia content. EX1003, ¶77. Hagai

³ A listing of challenged claims is provided in the Appendix.

describes known ways of building systems for transmission and playback, including a “data transmission apparatus,” or “HTTP server,” which transmits audio and video to a “data reception/playback apparatus,” or “playback device.” EX1004, 1:26-29, 1:53-58, Figs. 18-19. These systems could transmit multimedia content from servers over a network to devices for playback. EX1004, 2:56-58. However, these prior systems could not handle certain “playback control functions,” like playback from an arbitrary point in the video. EX1004, 3:1-13. Hagai explains how to handle these functions, including by using information associated with the multimedia content—metadata—to inform requests for playback from specific positions. EX1004, 4:13-32; EX1003, ¶77.

Li also relates to distributing and using multimedia files in a system where a server transmits multimedia over a network to a playback device. EX1005, Abstract, 1:18-22, 12:9-36, Fig. 1. As Li explains, it was known that “a device can seek to a particular portion of the multimedia information (e.g., a particular scene of a movie with a particular accompanying audio track . . .)” when decoding a multimedia file for playback and would have required accessing certain related information. EX1005, 51:60-64, 53:15-20; EX1003, ¶78. For example, the device may use “DMNU” chunks that contain descriptive metadata about media tracks, such as language options and references to information defining how the device may select corresponding audio and subtitle tracks. EX1005, 8:10-16, 8:53-59, 23:47-50,

24:57-61, Fig. 2.5, EX1003, ¶78. Further, Li describes DRM information stored within the multimedia file, such as that stored in “DRM chunks,” to decrypt encrypted video data for playback. EX1005, 27:42-28:15.

A POSITA would have been motivated to combine Li with Hagai such that the DMNU chunks of Li would be used by Hagai to provide the playback device with the necessary information regarding the stored audio tracks, video tracks, and subtitle tracks. EX1003, ¶¶ 76, 79.

A POSITA would have been further motivated to combine Hagai’s multimedia playback system with Li’s disclosure of how various additional metadata may be used for decoding and playback, including for DRM-protected multimedia. EX1003, ¶80. Hagai discloses using metadata during the video decoding and playback process, which was widely known. EX1003, ¶80. The combination would incorporate the additional metadata disclosed by Li in the same places where Hagai discloses such information may be stored. EX1003, ¶80. Since Hagai’s system already uses metadata associated with the media content during the decoding and playback process, it would have been straightforward to add Li’s functionality to take advantage of additional metadata used for decoding and playback of DRM-protected media sourced from the same places. EX1003, ¶80.

A POSITA would have had a reasonable expectation of success in making this combination because it merely adds information to the metadata affiliated with a

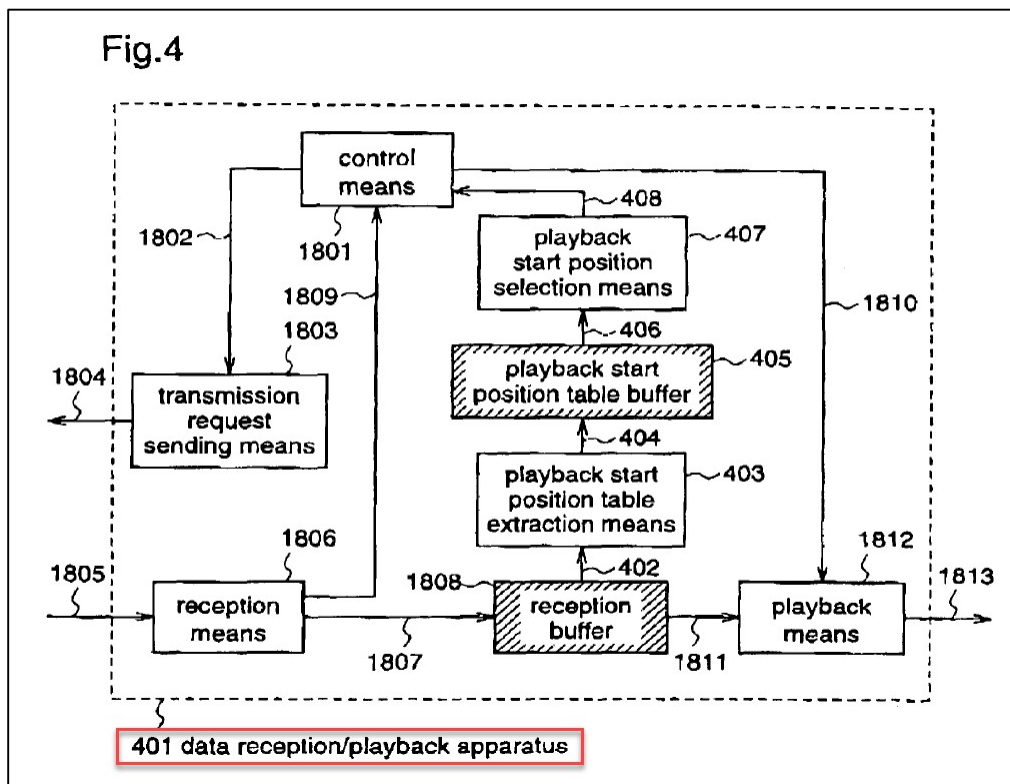
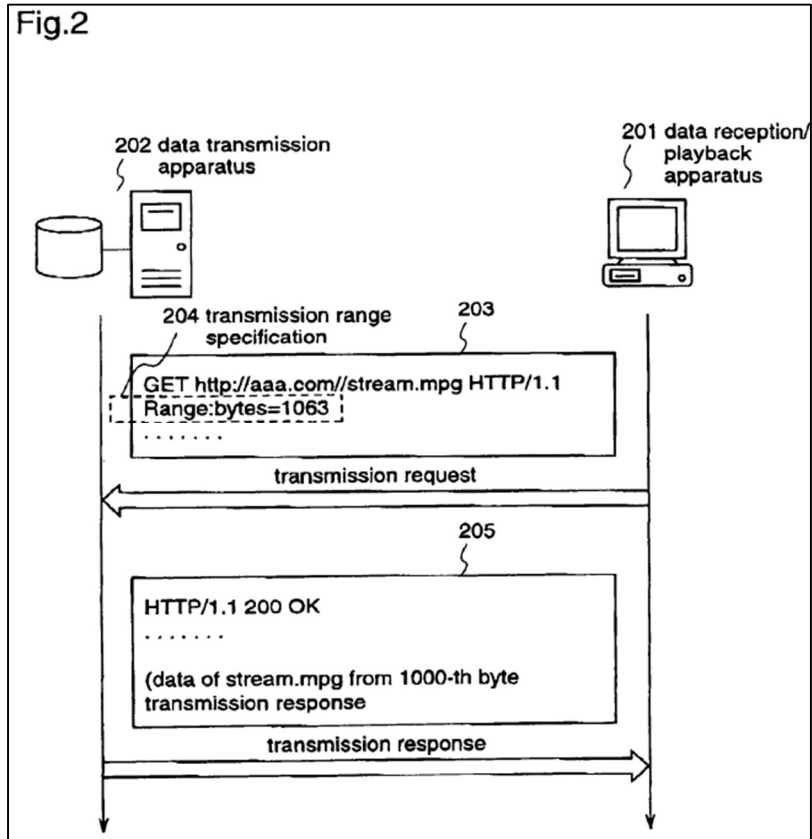
multimedia file and additional processing based on this information. EX1003, ¶81. The combination would have yielded predictable results because it is no more than a combination of conventional, well-known features in video transmission and playback systems. EX1003, ¶81.

B. Claim 1

1. [1.pre] “A playback device, comprising:”

To the extent the preamble of claim 1 is limiting, Hagai renders it obvious. EX1003, ¶82.

Hagai discloses a data reception/playback apparatus (“playback device”) that requests multimedia content from a data transmission apparatus 202 (“server”) using HTTP. EX1004, 2:56-58, 2:59-67, 9:4-10:10 (explaining that the playback apparatus “plays the stream data.”), Figs. 2, 4 (annotated); EX1003, ¶83.



Hagai provides examples of internet-connected playback devices, including “personal computers, mobile terminals, handy phones and the like.” EX1004, 1:17-20, Figs. 2, 4; EX1001, 6:10-20; EX1003, ¶84. A POSITA would have understood that these devices could be playback devices because they can play multimedia content received from a server. *Supra* § II.A.1.a; EX1003, ¶84.

2. [1.a] “a processor; and”

Hagai renders obvious limitation [1.a]. EX1003, ¶85.

Hagai discloses that the playback device comprises a “control means,” “transmission request sending means,” “reception means,” “playback means,” “playback start position table extraction means,” and “playback start position selection means.” EX1004, 9:1-14, Figs. 2, 4, 17-18; EX1003, ¶86. A POSITA would have understood that these “means” would be implemented by a processor because the functions they perform were conventional operations typically executed by processors in playback systems. EX1003, ¶86.

As explained above, Hagai discloses a playback device that can receive and play back multimedia content. EX1004, 9:4-10:10, 10:8-10, Figs. 2, 4; *supra* § IV.B.1; EX1003, ¶87. A POSITA would have understood that the playback device must include a processor to play back multimedia content. EX1003, ¶87. Furthermore, it was well-known to a POSITA that each of the exemplary playback

devices disclosed in Hagai would have a processor. EX1004, 1:17-20; *supra* §§ II.A.1.a, IV.B.1; EX1003, ¶87.

3. [1.b] “a non-volatile storage containing an application for causing the processor to perform the steps of:”

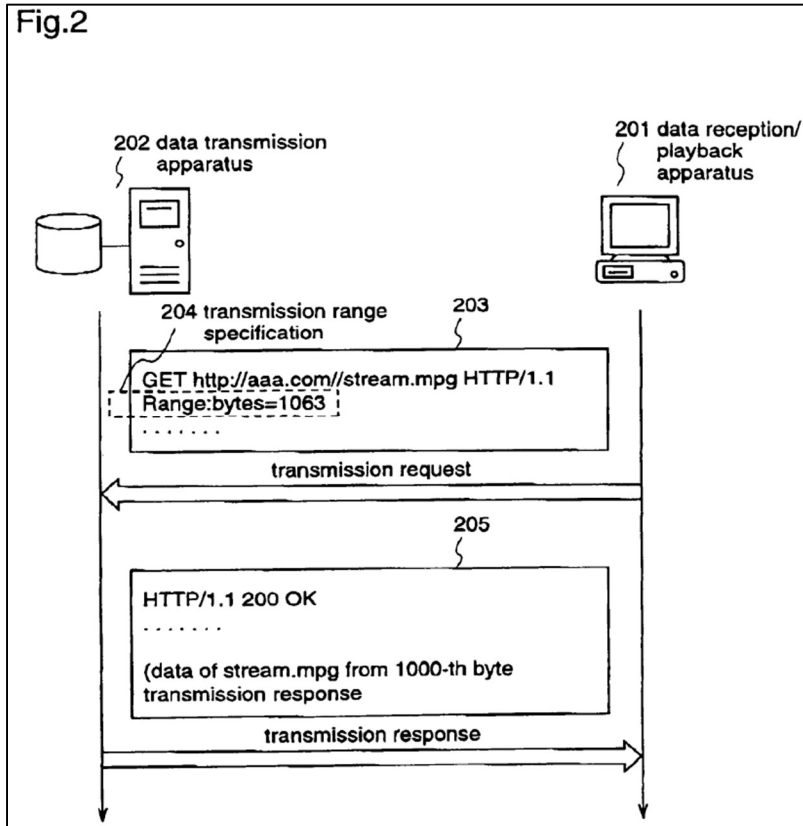
Hagai renders obvious limitation [1.b]. EX1003, ¶88.

As explained above, a POSITA would have understood that Hagai renders obvious a playback device with a processor. EX1004 at 1:17-20; EX1003, ¶89; *supra* §§ II.A.1.a, IV.B.1-IV.B.2. A POSITA would have known that such processors execute applications stored in a non-volatile storage to perform certain steps, including the ones explained below in Sections IV.B.4-IV.B.19. EX1003, ¶89. It was a well-known, routine practice to store playback applications in non-volatile storage to ensure operational readiness and data persistence. EX1003, ¶89.

4. [1.b(i)] “establishing at least one connection for communicating with a remote server system;”

Hagai discloses limitation [1.b(i)]. EX1003, ¶90.

Hagai discloses that the playback device communicates with the server using HTTP. EX1004, 8:19-27; EX1003, ¶91.



A POSITA would have understood that the playback device must establish a connection with a remote server system because the playback device transmits multiple data requests to, and receives data from, the server using an HTTP connection over the internet. EX1004, 1:16-24, 1:30-2:14; EX1003, ¶¶92-93. As discussed below in at least limitations [1.b(ii)], [1.b(iv)], [1.b(vi)], and [1.b(viii)], the playback device obtains a playback-speed/file mapping table (EX1004, 18:34-38, 18:52-62) and a playback start position table (EX1004, 8:41-48, 9:17-19) from the server, and transmits requests to the server for stream data from a start position. EX1004, 9:63-10:4; EX1003, ¶92; *infra* §§ IV.B.5, IV.B.7, IV.B.9, IV.B.11. A POSITA would have understood that these communications could not occur without

first establishing a connection between the playback device and server. EX1003, ¶92.

5. [1.b(ii)] “obtaining information from a remote server system describing at least one video track, multiple audio tracks, and multiple subtitle tracks;”

Hagai in combination with Li renders obvious limitation [1.b(ii)]. EX1003, ¶94.

Hagai discloses “obtaining information from a remote server system.” Hagai’s server transmits to the playback device a playback-speed/file mapping table before transmitting the multimedia file. EX1004, 18:34-38, 18:58-62; EX1003, ¶95.

This table describes multiple video tracks that can be transmitted from the server by identifying their playback speeds and the files containing them. EX1004, 18:38-45. For example, the file name “stream_s2.mpg” has a playback speed of “2.0,” indicating that the video and audio tracks of the “stream_s2.mpg” file have a playback speed double that of “normal playback speed.” EX1004, 18:14-20, 5:8-11, Fig. 16; EX1003, ¶96.

Fig.16

		1601	1602
		{	{
		playback speed	file name
1.0	stream.mpg		
2.0	stream_s2.mpg		
3.0	stream_s3.mpg		

A POSITA would have understood that the playback-speed/file mapping table includes information describing at least one video track and multiple audio tracks because it provides the speed at which the playback device plays the video and/or audio tracks of each multimedia file (e.g., stream_s2.mpg) available from the server. EX1003, ¶97.

To the extent that Hagai does not disclose information describing at least one video track, multiple audio tracks, and multiple subtitle tracks, Li does. EX1003, ¶98. Li discloses a multimedia file that includes “PlayAction” chunks 182, which describe available tracks using references to a “MediaTrack” chunk 198. EX1005, 24:2-9, Fig. 2.5 (annotated below).

Li discloses a multimedia file stored on a server structured as a RIFF file. EX1005, 51:50-58, 12:46-53; EX1003, ¶¶99-100. Li further discloses that the media file contains “DMNU” chunks (EX1005, 14:62-67), including various menu chunk objects and references to media tracks. EX1005, 23:13-15, 23:30-35, Fig. 2.5. These chunks include descriptive metadata about media tracks, such as language options and references to media chunks—information defining how the device may select corresponding tracks. EX1005, 8:10-16, 8:53-59, 23:47-50, 24:57-61, Fig. 2.5, EX1003, ¶101.

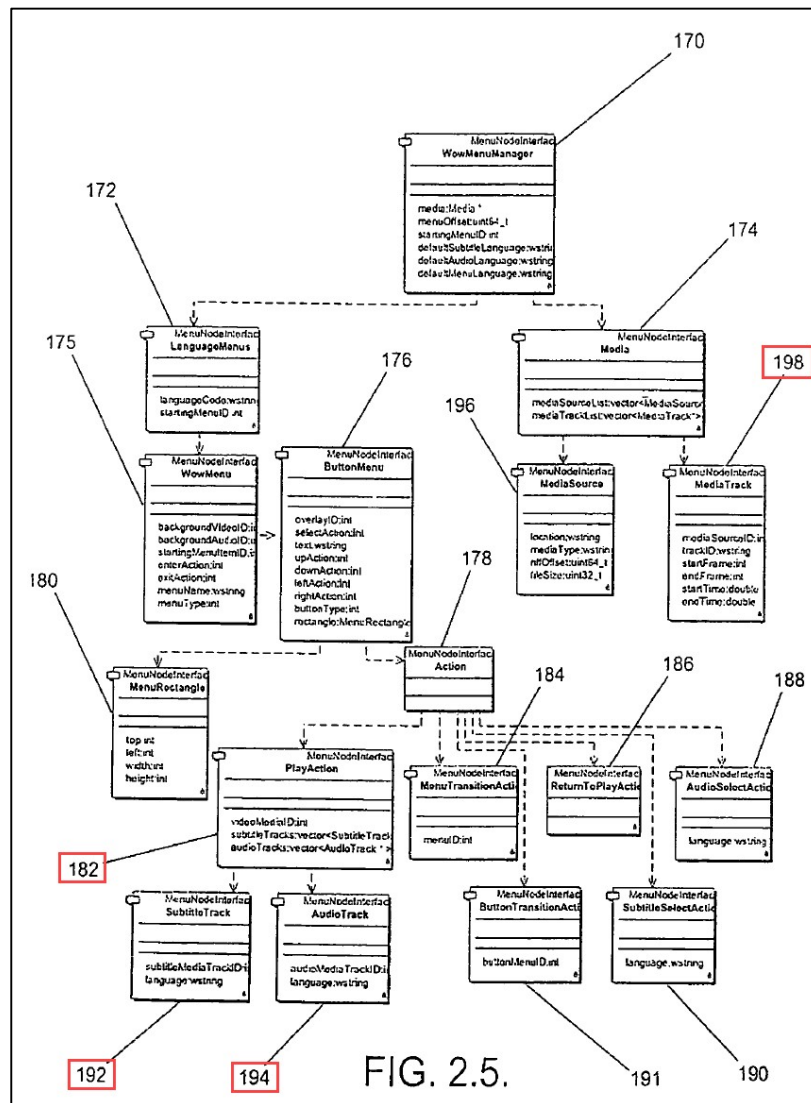


FIG. 2.5.

One of Li's menu chunk objects is the "PlayAction" chunk 182, which identifies the video, audio, and subtitle tracks within the multimedia file, and identifies the corresponding audio and subtitle tracks "using 'SubtitleTrack' 192 and 'AudioTrack' 194 chunks." EX1005, 23:58-24:2, Fig. 2.5 (annotated); EX1003, ¶¶102-04. The SubtitleTrack and AudioTrack chunks contain references to a "MediaTrack" chunk 198 that allow selection of, for example, audio and subtitle tracks for a specific video track. EX1005, 24:2-4, 24:43-45, 25:27-33.

A POSITA would have understood that the SubtitleTrack chunk 192 and AudioTrack chunk 194 describe multiple audio tracks and multiple subtitle tracks because they identify the tracks available for selection by the playback device and provide information necessary to select them in response to user preferences. EX1005, 23:67-24:2, 24:19-31, Fig. 2.1; EX1003, ¶105. Indeed, the 141 patent refers to Li's menu chunks as providing the descriptive information necessary for track selection. EX1001, 9:49-53, 10:14-27; EX1003, ¶106.

A POSITA would have been motivated to store on the server of Hagai the DMNU chunks of Li, including the SubtitleTrack and AudioTrack chunks. EX1003, ¶107. Hagai discloses that the server stores a multimedia file and a playback-speed/file mapping table that describes available tracks and their respective playback speeds. EX1004, 17:55-60, 18:51-59; EX1003, ¶107. As Li explains, the SubtitleTrack and AudioTrack chunks also reference MediaTracks, so a POSITA would know to relate them to their corresponding video tracks in Hagai. EX1005, 23:3-24:10, 24:19-31, Figs. 2.1. 2.5; EX1003, ¶107. Thus, in the combined configuration, the server of Hagai would store the multimedia file and playback-speed/file mapping table and the DMNU chunks describing available tracks and providing information necessary to select them. EX1003, ¶107. A POSITA would have understood that modifying the server to store the DMNU chunk of Li, which includes the SubtitleTrack and AudioTrack chunks, is a combination of prior art

elements according to known methods to achieve predictable results that would involve storing metadata chunks, in addition to the playback-speed/file mapping table and the multimedia file of Hagai. EX1004, 1:29-45, 2:19-25; EX1003, ¶107.

A POSITA would also have been motivated to modify Hagai such that the DMNU chunks of Li would be transmitted from the server to the playback device with the playback-speed/file mapping table before transmission of the streaming data file. EX1003, ¶108. A POSITA would have understood that the playback device must have the necessary information regarding the stored audio tracks, video tracks, and subtitle tracks before they could be selected for playback. EX1003, ¶108. This track information is necessary for the playback device to select the media data to request. EX1003, ¶108.

This is a combination of prior art elements according to known methods to yield predictable results, as well as use of a known technique to improve a similar method in the same way. EX1003, ¶109. As noted above, because Hagai's server already stores the playback-speed/file mapping table and multimedia file, and transmits the table to the playback device, modifying Hagai such that the server stores Li's metadata chunks and transmitting such information to the playback device would be a reasonable modification that provides expected results. EX1003, ¶109.

6. [1.b(iii)] “selecting a video track from the at least one video track;”

Hagai discloses limitation [1.b(iii)]. EX1003, ¶110.

Hagai discloses that the playback-speed/file mapping table contains information about video tracks with different playback speeds and their corresponding files. EX1004, 18:14-20, Fig. 16. Based on the information in the table, the playback device selects a video track to request from the server. EX1004, 18:38-44. In response, the server determines the corresponding file containing the selected video track and transmits it to the device, including in response to byte range requests. EX1004, 18:38-50; *infra* IV.B.11. A POSITA would have understood that Hagai discloses “selecting a video track.” EX1003, ¶111.

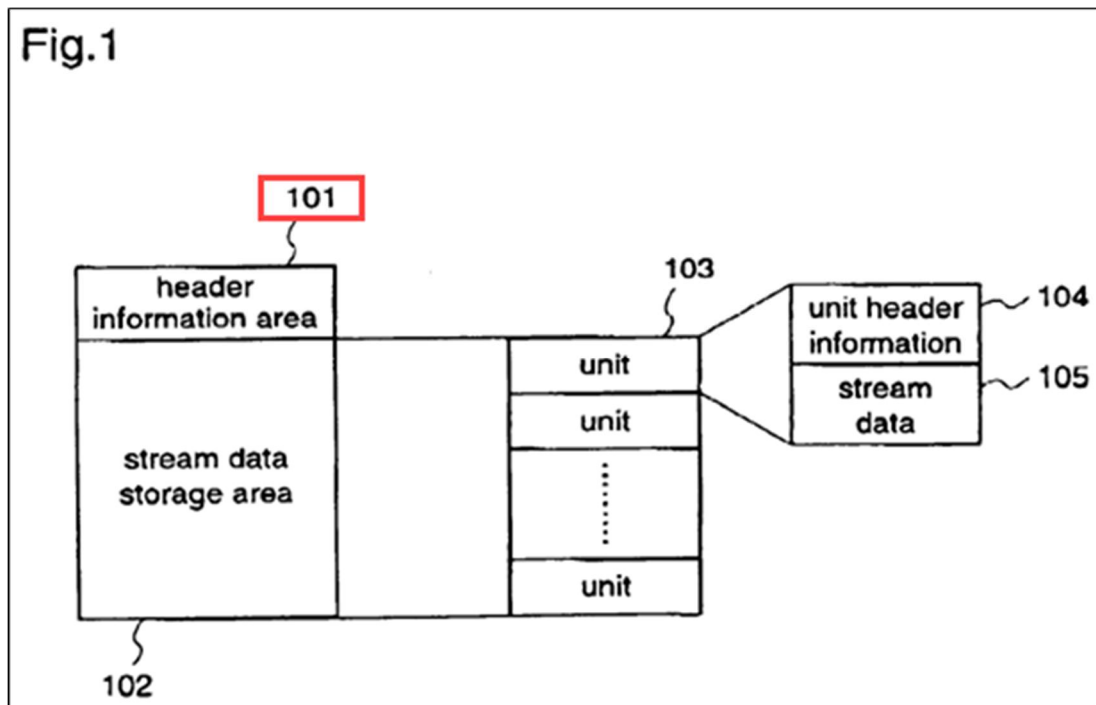
7. [1.b(iv)] “requesting a header describing the selected video track;”

Hagai discloses limitation [1.b(iv)]. EX1003, ¶112.

Hagai discloses selecting a file for playback and requesting the associated stream data. EX1004, 8:21-36, 18:58-67. A POSITA would have understood that when the playback device sends an initial transmission request to the server for a portion of stream data starting from the head position of the multimedia file, the data would include the header information area 101, which is the first part of the file. EX1004, 8:21-29, 14:44-48, 18:58-67, 19:51-55; *see also id.*, 9:19-23; EX1003, ¶¶113-14. A POSITA would have understood that the request would include the

header information area 101 to enable the playback device to transmit requests for byte ranges starting from an arbitrary part of the multimedia file. EX1004, 8:41-65, 9:63-10:34; EX1003, ¶114.

Hagai further discloses that the header information 101 describes the selected video track. Figure 1 of Hagai discloses the general file format of stream data, which includes header information area 101 and a stream data storage area 102 that comprises a plurality of units 103. EX1004, 8:15-18, 10:35-41, Fig. 1 (annotated).



A POSITA would have understood that the header information unit 101 describes the selected track because the header information area 101 provides metadata describing the type of media included in the multimedia file. EX1004, 8:16-18; EX1003, ¶¶115-17.

8. [1.b(v)] “selecting an audio track from the multiple audio tracks;”

Hagai in combination with Li discloses limitation [1.b(v)]. EX1003, ¶118.

To the extent that Hagai does not expressly disclose limitation [1.b(v)], Li does. EX1003, ¶119. Li discloses that the media file contains “DMNU” chunks, including various menu chunk objects and references to media tracks. *Supra* § IV.B.5. These chunks include descriptive metadata defining how the device may select corresponding tracks. *Id.* Li further discloses an “AudioSelectAction” chunk that is part of the DMNU chunk. EX1005, 24:19-20. This AudioSelectAction chunk contains “information that can be used to select a particular audio track,” and allows the playback device to locate and select audio tracks, which may be contained in a multimedia file or in an externally referenced file. EX1005, 23:3-24:10, 24:19-24, Fig. 2.5; *supra* § IV.B.5.

A POSITA would have been motivated to store Li’s DMNU chunk on Hagai’s server and use the AudioSelectAction chunk to select an audio track. EX1003, ¶¶119-120; *supra* § IV.B.5. For example, in the combined configuration, the server of Hagai would store the multimedia file and playback-speed/file mapping table of Hagai, and the DMNU chunk of Li (containing the AudioSelectAction chunk). EX1003, ¶120; *supra* § IV.B.5. A POSITA would have understood that modifying the server to store the DMNU chunk of Li is a combination of prior art elements according to known methods to achieve predictable results and would involve

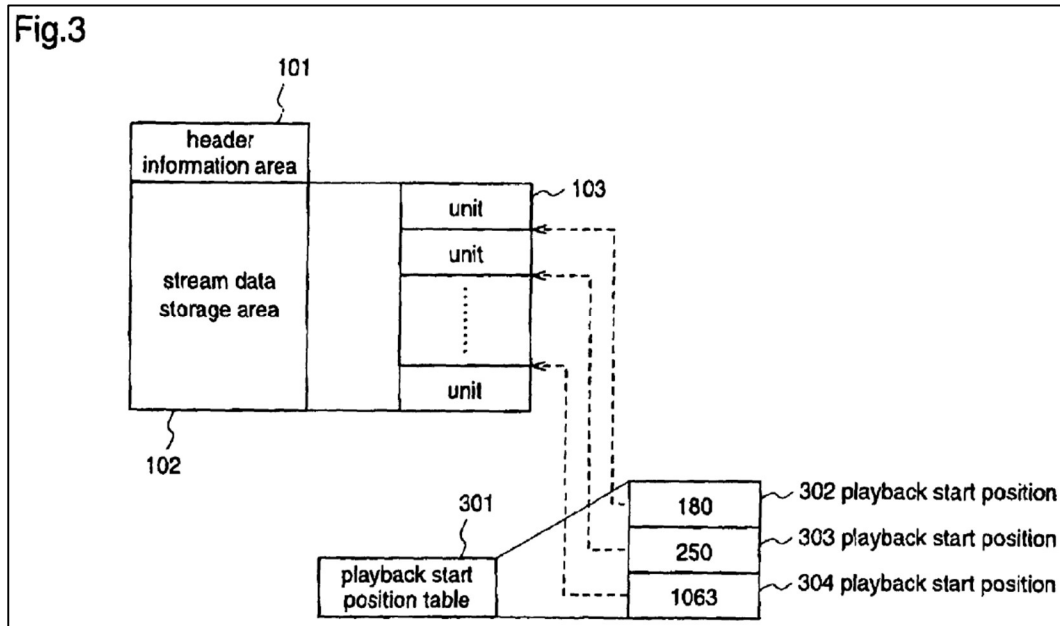
storing metadata chunks, in addition to the playback-speed/file mapping table and Hagai's multimedia file, on Hagai's server. EX1004, 1:29-45, 2:19-25; EX1005, 23:67-24:2, 24:19-31, Fig. 2.1; *supra* § IV.B.5; EX1003, ¶120. Moreover, Hagai discloses the selection of a video track, and Li simply provides another mechanism, via the AudioSelectAction chunk, to select an audio track. EX1005, 24:20-24; *supra* § IV.B.6. Thus, modifying Hagai such that the server stores Li's DMNU chunk, including the AudioSelectAction chunk, would be a reasonable modification that provides expected results. EX1003, ¶120.

9. [1.b(vi)] “obtaining index information indicating the locations of audio and video data within the selected audio and video tracks;”

Hagai in combination with Li renders obvious limitation [1.b(vi)]. EX1003, ¶121.

Hagai discloses obtaining index information indicating the locations of audio and video within the selected audio and video tracks. EX1003, ¶122. As discussed in limitation [1.b(iv)], Hagai discloses a general file format for stream data in Figure 1 along with a “playback start position table” containing start positions enabling playback of the video and/or audio content from arbitrary points in the file. EX1004, 8:11-48, Fig. 1. As illustrated in Figure 3, the playback start position table 301 indexes playback start positions 302, 303, and 304, corresponding to “the head

positions of units constituting the stream data storage areas.” EX1004, 8:55-59, Fig. 3; EX1003, ¶123.



The playback start position table further includes “the number of bytes from the head position of the stream data storage area 102” that enables the playback device to “specify the unit head position even when” it receives data from an arbitrary point in the file. EX1004, 8:49-59; EX1003, ¶124.

A POSITA would have understood that the playback start position table 301 is index information indicating the locations of audio and video data within the selected audio and video tracks. The playback start positions indicate “the positions in the file where data playback is possible,” allowing the playback device to play the selected tracks even when playback begins from an arbitrary location in the stream,

such as by selection of the location by a user in a jump playback action. EX1004, 8:36-65; EX1003, ¶125.

A POSITA would have understood this applies to both audio and video tracks. Hagai discloses that selection “enables playback of audio/video contents from an arbitrary position in stream data,” and that the stream data includes both audio and video. EX1004, 8:4-8; *see also, id.* 1:9-12, 1:53-58, Cls. 1, 7-11. A POSITA would have understood that each unit would contain both video and its corresponding audio, such as by interleaving the audio with the video within each unit. EX1004, 8:15-28 (disclosing that the header describes the different “types of media stored in the respective units of the stream data storage area”); EX1003, ¶126.

To the extent this is not expressly disclosed by Hagai, it is obvious in view of Li. *Id.* Li discloses that its RIFF file structure stores “video” chunks and interleaved “audio” chunks within the same “RIFF” chunk. A POSITA would have understood that this file structure was well-known and would permit Hagai’s unit-based approach to achieve its disclosed result. *Id.* If a POSITA needed explicit disclosures or teaching of this type of file structure, a POSITA would be motivated to look to Li. *Id.*

A POSITA would further understand that the playback device “obtains” the playback start position table from the server because the playback device requests

and receives the playback start position table separately from the stream data. EX1004, 9:24-29; *see also id.*, 4:60-67, 8:36-48, 10:11-34, 19:51-59; EX1003, ¶127.

10. [1.b(vii)] “determining byte ranges to request from the selected audio and video tracks using the index information;”

Hagai discloses limitation [1.b(vii)]. EX1003, ¶128.

Hagai discloses that the playback start position selection means “selects” a playback start position from the playback start position table. EX1004, 9:37-46, Fig. 4; *supra* § IV.B.9. Using the information in the table, the playback device determines the location of the selected start position within the file. EX1004, 8:55-65; *supra* § IV.B.9. The playback start positions indicate the head positions of units constituting the stream data storage area, which contains both audio and video, and each playback start position indicates the byte offset from the beginning of the stream data to the start position. EX1004, 8:4-10, 55-65; EX1003, ¶129.

Based on the selected playback start position, the playback device determines a byte range to request that starts with the byte identified in the start position table and terminates at the next playback start position of the next portion of video. EX1004, 8:60-65, 9:63-10:4; EX1003, ¶130; *infra* § IV.B.11. A POSITA would have understood that by determining a subset of the file to request using a selected position in the start position table, Hagai’s system is “determining byte ranges to request from the selected audio and video tracks using the index information.” EX1003, ¶130.

11. [1.b(viii)] “requesting byte ranges from the selected video track and the selected audio track from the remote server system;”

Hagai in combination with Li renders obvious limitation [1.b(viii)]. EX1003, ¶131.

Hagai discloses that after selecting the playback start position, the transmission request sending means of the playback device sends an HTTP transmission request to the server for a specified transmission range. EX1004, 9:63-10:4. The transmission range identifies a transmission start position that “indicates the number of bytes from the head of the file” to the selected start position. EX1004, 8:27-29, 8:55-59. HTTP requests were well-known in the art, and a POSITA would have understood that it was conventional to request byte ranges including both a start and end point—especially to request a subset of data from large files such as those containing videos. EX1003, ¶132; *supra* § II.A.1.a. Thus, a POSITA would have understood it to be obvious that “transmission range 204” identifies the start of the byte range, which is requested by the playback device using HTTP. EX1004, 8:19-29, 8:55-65, 9:41-62, Fig. 2; EX1003, ¶132.

A POSITA would have understood that the transmission request is a request for the byte range of the “selected video track and the selected audio track.” EX1003, ¶133. The request represents the byte range between the selected transmission start position and a terminus, like the next playback start position in the

stream data storage area, which contains both video and audio from the selected tracks. EX1004, 8:4-10, 27-29, 55-59, 9:63-10:21; EX1003, ¶133; *supra* § IV.B.10.

To the extent Hagai does not expressly disclose this limitation, it is rendered obvious in view of Li. EX1003, ¶133. Li describes how its RIFF file structure stores video chunks and interleaved audio chunks within the same RIFF chunk. EX1005, 1:54-65. A POSITA would have been motivated to combine this teaching with the unit-based approach of Hagai to ensure that its requests based on index information request byte ranges of the audio and video tracks. *Supra* § IV.B.9.

12. [1.b(ix)] “buffering received bytes of information comprising audio and video data;”

Hagai discloses limitation [1.b(ix)]. EX1003, ¶134.

After sending the byte range request to the server, the playback device receives the requested data and stores it “in the reception buffer.” EX1004, 10:4-7, Fig. 2. Hagai further discloses that the requested stream data includes both audio and video data. EX1004, 2:56-58, 5:7-10. Thus, a POSITA would have understood Hagai to disclose receiving and buffering bytes of audio and video data in preparation for playback. EX1003, ¶135.

13. [1.b(x)] “checking that sufficient data is buffered to commence playback and playing back the buffered audio and video data;”

Hagai discloses limitation [1.b(x)]. EX1003, ¶136.

Hagai’s playback device monitors whether the amount of buffered data is sufficient before commencing or resuming playback. The playback device includes “playback means for playing the stream data stored in the reception buffer” and “reception buffer monitor means for monitoring the quantity of data stored in the reception buffer.” EX1004, 6:10-14, 16:48-51. As explained above, although Hagai discloses five embodiments, a POSITA would have understood that these embodiments are compatible modules that can be combined (e.g., the first and third embodiments). EX1003, ¶137; *supra* § III.A. Playback is suspended when the buffer level drops below a threshold. EX1004, 6:15-27, 16:61-67, 17:34-41. When the buffer monitor detects that the amount of buffered data exceeds a predetermined value, playback restarts. EX1004, 6:21-27. A POSITA would have understood that the playback device would check for sufficient buffered data before commencing playback. EX1003, ¶138.

14. [1.b(xi)] “responding to a received seek instruction by:”

Hagai discloses limitation [1.b(xi)]. EX1003, ¶139.

Hagai describes a “jump playback function,” which allows playback to begin “from an arbitrary position in stream data.” EX1004, 8:4-10. For example, the

playback device can respond by jumping to a playback time (e.g. 3 seconds) indicated by the user. EX1004, 9:47-54. A POSITA would have understood the jump playback function as corresponding to a seek instruction. EX1003, ¶140. Hagai discloses responding to the instruction, as explained below.

15. [1.b(xi)(1)] “pausing playback;”

Hagai discloses limitation [1.b(xi)(1)]. EX1003, ¶141.

As discussed above, Hagai discloses that data may be played from an arbitrary point in response to a user request (i.e., a seek instruction). EX1004, 9:47-54; *supra* § IV.B.14. Hagai explains that when this is the case, the playback device temporarily halts data reception and playback to determine the new transmission start position. EX1004, 11:20-25. Hagai explains that “data reception is halted temporarily” during this process, and playback resumes from the selected playback start position once the new start point is determined. EX1004, 11:20-25, 13:47-56; EX1003, ¶142.

16. [1.b(xi)(2)] “determining byte ranges to request from the selected audio and video tracks based upon a new playback location using the index information;”
17. [1.b(xi)(3)] “requesting byte ranges required to play the selected audio and video tracks from the new playback location from the remote server;”
18. [1.b(xi)(4)] “buffering received bytes of information comprising audio and video data pending resumption of playback; and”
19. [1.b(xi)(5)] “checking that sufficient data is buffered to commence playback and playing back the buffered audio and video data.”

Upon receiving a seek instruction, Hagai’s playback device repeats the same sequence of operations used for playback before the seek instruction. These operations recited in [1.b(xi)(2)]-[1.b(xi)(5)] are identical to limitations [1.b(vii)]-[1.b(x)] above, except that they are performed in relation to the new playback location based on the seek instruction received in limitation [1.b(xi)]. EX1003, ¶143. These elements are obvious for the same reasons explained with respect to [1.b(vii)]-[1.b(x)].

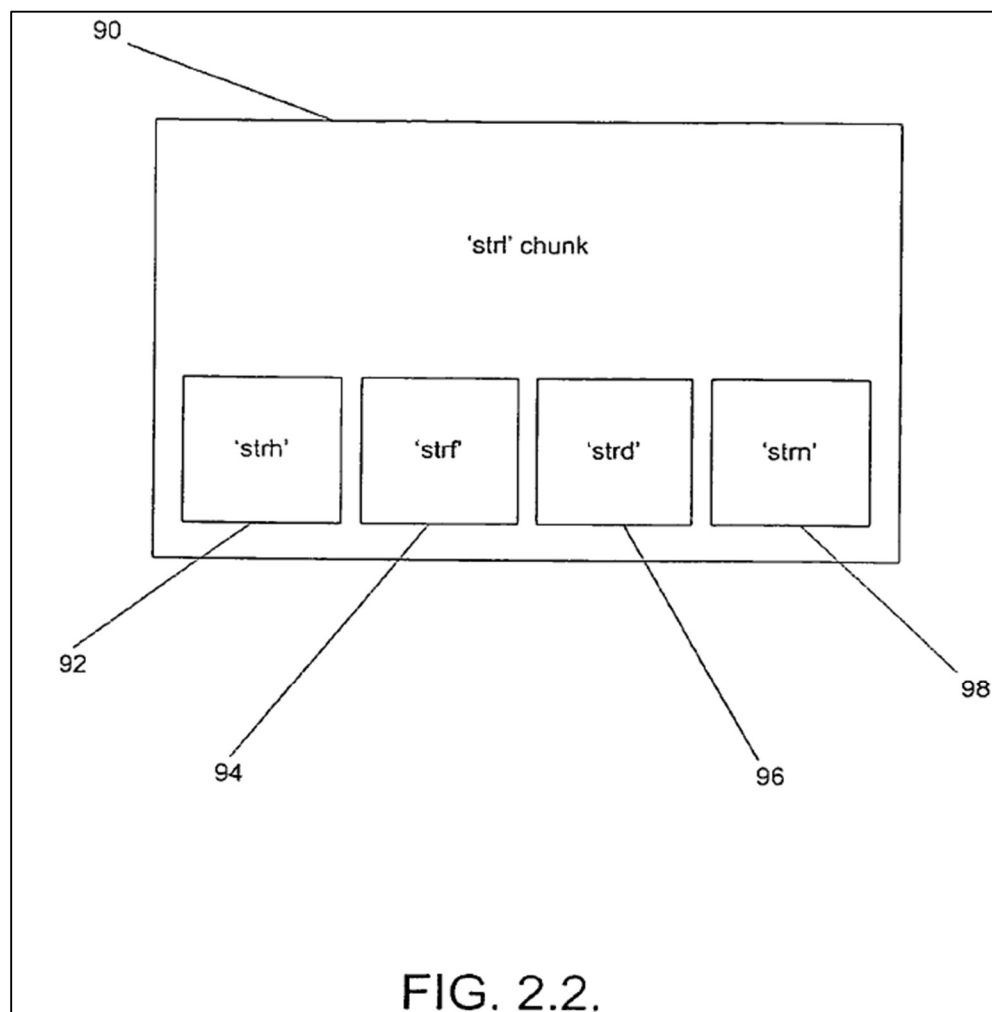
C. Claim 3: “The playback device of claim 1, wherein the step of requesting a header describing the selected video track comprises the step of requesting a DRM header.”

Hagai combined with Li renders obvious claim 3. EX1003, ¶144.

A POSITA would have understood that when the playback device of Hagai sends a request to the server for a header describing the selected video track, the request would also include a request for the DRM header. *Supra* § IV.B.7; EX1003,

¶145. It was conventional to encrypt multimedia content to prevent unauthorized playback. EX1003, ¶145; *supra* § II.A.1.d. A POSITA would have understood that when encrypting content, the encrypting device generates a decryption key, which is stored or referenced in the DRM header. EX1005, 27:64-28:15, 52:41-55; EX1003, ¶145; *supra* § II.A.1.d.

Li discloses that the multimedia file includes a “strd” chunk, as illustrated in Fig. 2.2, which indicates whether the content is protected by DRM and includes a DRM header. EX1005, 14:40-44, 27:64-28:2, Fig. 2.2; EX1003, ¶¶146-148.



The DRM header includes encryption keys to decrypt chunks of multimedia content stored in the multimedia file. EX1005, 27:64-28:2, 52:44-55. Because the encryption keys in the DRM header are necessary for decryption, the DRM header must be requested and received before decoding and playback. EX1005, 52:49-55, Fig. 4. A POSITA would have understood that when the playback device requests a header from the server, the playback device would also request the DRM header of Li to access the keys required to decrypt the encrypted content of the multimedia file. EX1005, 27:64-28:4, 52:41-55; *supra* § IV.B.7; EX1003, ¶149.

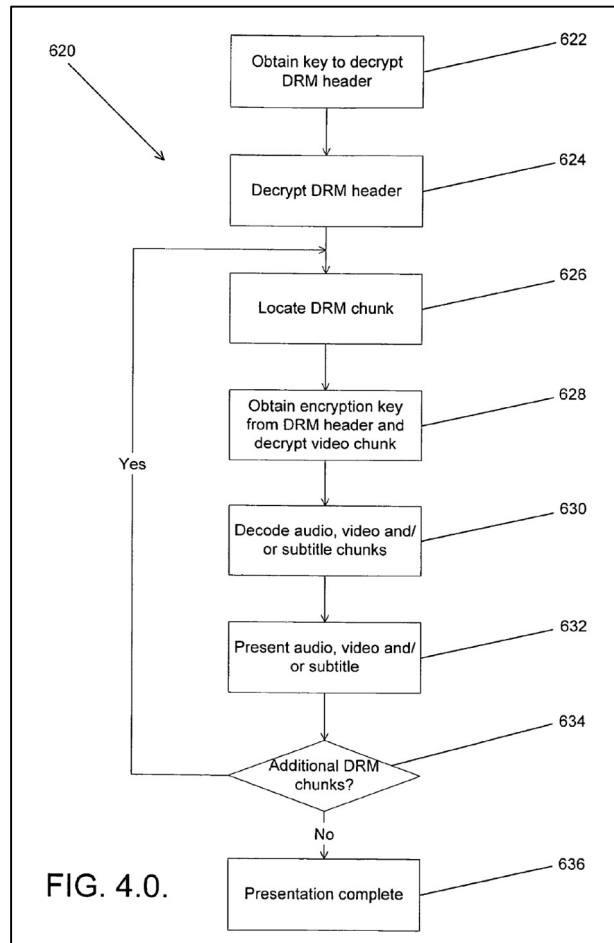
A POSITA would have found it obvious to store the DRM header on Hagai's server. EX1003, ¶150. As explained in Section IV.B.5, Hagai discloses that the server stores the multimedia file and a playback-speed/file mapping table. *Supra* § IV.B.5; EX1003, ¶150. Thus, in the combined configuration, the server of Hagai would store the multimedia file and playback-speed/file mapping table as well as the DRM header corresponding to the tracks stored on the server. EX1003, ¶150. A POSITA would have understood that modifying the server to store the DRM header of Li is a combination of prior art elements according to known methods to achieve predictable results and would involve saving data chunks, in addition to Hagai's playback-speed/file mapping table and the multimedia file, on Hagai's server. EX1004, 1:29-45, 2:19-25; EX1003, ¶150.

A POSITA would be motivated to modify Hagai such that the DRM header of Li is transmitted with Hagai's header information. EX1003, ¶151. A POSITA would have understood that requesting and receiving Li's DRM header alongside Hagai's header information area, as explained above, would enable the playback device to decrypt received data without the delay caused by additional requests. EX1004, 8:15-18; *supra* § IV.B.7; EX1003, ¶151. Indeed, this modification of the playback device of Hagai is a combination of known methods to achieve predictable results because the combined request from the playback device merely adds a request for the DRM header to Hagai's existing request for header information stored on the server. EX1004, 1:29-45, 2:19-25; *supra* § IV.B.7; EX1003, ¶151.

D. Claim 4: “The playback device of claim 3, wherein the application is further capable of causing the processor to perform the step of decrypting the DRM header.”

Hagai combined with Li renders obvious claim 4. EX1003, ¶152.

Li discloses in Figure 4 a process for “locating the required multimedia information from a multimedia file.” EX1005, 52:41-44. The process includes obtaining a key to decrypt the DRM header in step 622 and subsequently decrypting the DRM header in step 624 using the encryption key. EX1005, 52:44-55, Fig. 4; EX1003, ¶153.



A POSITA would have understood that in the combined configuration, once the playback device requests and receives the DRM header, the playback device would obtain the encryption key and decrypt the DRM header. EX1005, 52:41-55; *supra* § IV.C; EX1003, ¶154. A POSITA would have understood that decryption techniques were conventional and well-known, and that the processor of the playback device could perform these decryption techniques using known methods. EX1003, ¶154; *supra* § II.A.1.d.

As discussed in Section IV.C, a POSITA would have been motivated to combine Li's decryption functionality with Hagai's multimedia playback system to

enable playback of protected content using standard DRM techniques. EX1003, ¶155. A POSITA would have been further motivated to modify Hagai’s multimedia playback system to include Li’s DRM header decryption because the DRM header would have to be encrypted to secure the decryption keys within. EX1005, 27:67-28:4; *infra* § IV.E.1. Incorporating Li’s DRM header processing into Hagai would have yielded predictable results, as it merely adds known decryption functionality without altering Hagai’s core operation. EX1003, ¶155; *supra* § IV.C.

E. Claim 5

- 1. [5.pre], [5.a] “The playback device of claim 3,
wherein: the DRM header identifies at least one key;
and”**

Hagai combined with Li discloses limitations [5.pre] and [5.a]. EX1003, ¶156.

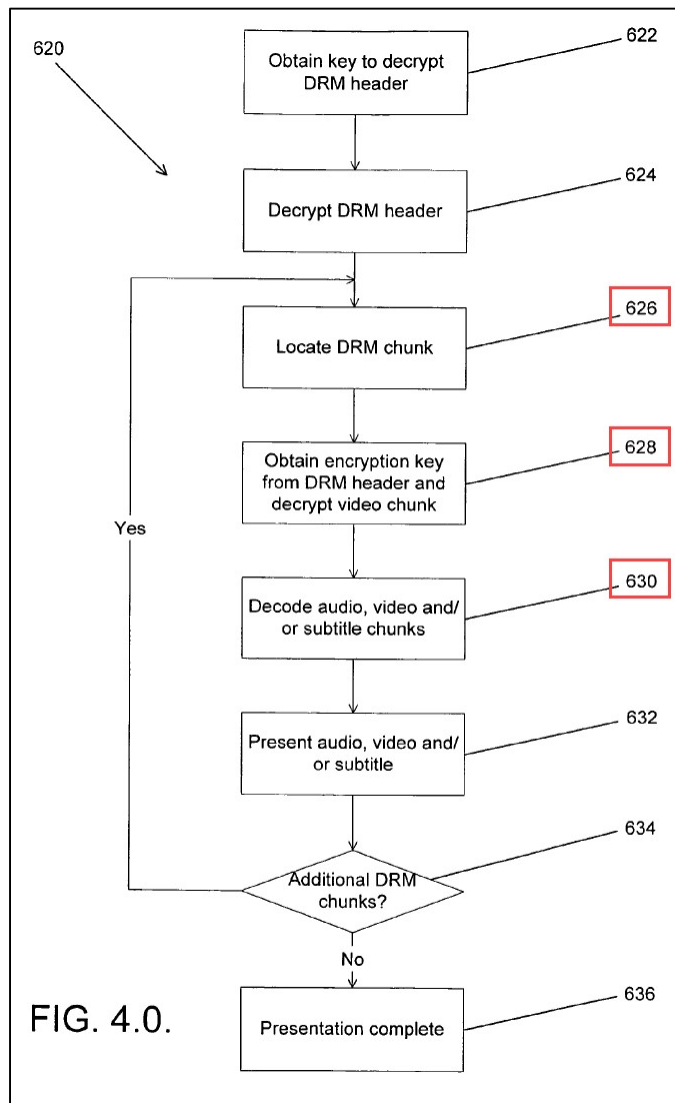
Li discloses that the DRM header contains a table of decryption keys (EX1005, 52:49-51), which are encrypted using a master key. EX1005, 28:2-3. The DRM header “contains information identifying the master key.” *Id.* 28:3-4. A POSITA would have been motivated to store Li’s DRM header on the server of Hagai for the reasons discussed above. *Supra* §§ IV.C, IV.D; EX1003, ¶157.

2. **[5.b] “the application is further capable of causing the processor to perform the steps of decrypting encrypted frames of video using the at least one key prior to decoding.”**

Hagai combined with Li discloses limitation [5.b]. EX1003, ¶158.

As discussed above, the DRM header identifies the master key. *Supra* § IV.E.1. Li explains that the master key is used to scramble and encrypt the decryption keys. EX1005, 27:67-28:4. A POSITA would have understood that the same master key is used to decrypt the decryption keys, which are then used to decrypt the frames of video. *Id.*; EX1003, ¶159.

Specifically, the process includes locating the DRM chunk of the multimedia file in step 626, obtaining an encryption key (i.e., an encryption key decrypted using the master key) from the DRM header to decrypt an encrypted video chunk in step 628, and then decoding the video chunk in step 630. EX1005, 52:49-55, Fig. 4 (annotated); EX1003, ¶160.



A POSITA would have understood that the master key used to decrypt the encryption key obtained from the DRM header at step 628 is a key used by the processor to perform the process of decrypting encrypted frames of video prior to decoding. EX1003, ¶161. In other words, the application uses the master key to enable each of the keys to decrypt the individual video chunks in step 628. *Id.*

A POSITA would have been motivated to combine these disclosures from Li with Hagai because it would be understood that the playback device must first

decrypt the encoded video frames using a key before decoding; the playback device cannot decode and play back encrypted video data, and Li teaches a secure way of providing that functionality. EX1003, ¶162; *supra* § II.A.1.d. Further, a POSITA would have had a reasonable expectation of success, and this combination would have yielded predictable results because it is no more than a combination of using conventional, well-known decryption techniques in a conventional way in video transmission and playback system. EX1003, ¶162.

F. Claim 6

- 1. [6.pre], [6.a] “The playback device of claim 5, wherein: encrypted frames of video are only partially encrypted;”**

Hagai combined with Li discloses limitations [6.pre] and [6.a]. EX1003, ¶163.

Li discloses that the multimedia file includes “a series of encoded video frames” that “are encoded as a series of video chunks.” EX1005, 4:42-45, Abstract. Li further discloses that the video chunks of the multimedia file could be partially encrypted. EX1005, 27:62-64. Because video frames may be encoded as video chunks, a POSITA would have understood that partially encrypted video chunks may include partially encrypted frames. EX1005, 4:42-45 (“the series of encoded video frames are encoded as a series of video chunks”), 3:17-23 (“each chunk of video information can include a single frame of video”), 3:60-63 (“a single frame of

video is contained within each ‘video’ chunk”); EX1003, ¶164. Li further explains that DRM chunks of the multimedia file provide DRM information for the video chunk and include information “point[ing] to the start of the encrypted block within the frame.” EX1005, 28:10-13, 27:53-56; *infra* § IV.F.2. Furthermore, it was well-known to a POSITA to encrypt only portions of video frames to reduce processing overhead while maintaining sufficient protection of digital content. EX1003, ¶165; *supra* II.A.1.d. Thus, a POSITA would have understood that Li discloses a system where only specific portions of the video frames—such as selected blocks within a frame—are encrypted. EX1003, ¶165.

As discussed with respect to claims 3-5, a POSITA would have been motivated to incorporate Li’s partial encryption scheme into Hagai’s multimedia playback system. EX1003, ¶166. The combination would have yielded predictable results, enabling secure playback of protected content using well-known partial encryption techniques without requiring changes to Hagai’s existing byte-range-based retrieval and playback architecture. *Id.*

Partial encryption was well-known to a POSITA before January 2007. EX1003, ¶167; *supra* § II.A.1.d. Indeed, the specification of the 141 patent does not disclose any novel partial encryption or decryption technique but instead incorporates prior art by reference—including Li, which provides a detailed discussion of partial encryption. EX1003, ¶167.

2. [6.b] “the received bytes of information contain references to portions of video frames that are encrypted; and”

Hagai combined with Li discloses limitation [6.b]. EX1003, ¶168.

A POSITA would have understood that the DRM chunks of Li, as explained in limitation [6.a], include references to portions of video frames that are encrypted. EX1003, ¶169. Li discloses that DRM chunks contain information needed to decrypt and decode media content. EX1005, 27:48-59. The DRM chunk includes an offset value, which specifies “the start of the encrypted block within the frame,” and a number value that “indicates the number of encrypted bytes in the block.” EX1005, 28:10-15, Fig. 2.9; EX1003, ¶170. A POSITA would have understood that the received DRM chunk includes offsets and byte ranges within frames by explicitly identifying the regions of the video frames that are encrypted, thereby enabling the playback device to locate and decrypt the encrypted portions of partially encrypted frames. EX1003, ¶171.

3. [6.c] the step of decrypting encrypted frames of video using the at least one key prior to decoding further comprises decrypting encrypted portions of frames of video using the references to the portions of the video frames that are encrypted.”

Hagai combined with Li discloses limitation [6.c]. EX1003, ¶172.

As discussed in limitation [5.b], Li discloses that the DRM chunk includes a key value 288, which “references the decryption key that can be used to decrypt” the

encrypted portions of the frame. EX1005, 28:10-15, Fig. 2.9; *supra* § IV.E.2. A POSITA would have understood that the key value 288 identifies the key needed to decrypt the encrypted portions of the frame, which are located using the information in the DRM chunk. EX1005, 28:10-15, Fig. 2.9; *supra* § IV.F.2; EX1003, ¶173. Thus, decrypting the encrypted portions of frames includes “using the references to the portions of the video frames that are encrypted.” EX1005, 28:10-15, Fig. 2.9; EX1003, ¶173.

As discussed with respect to claims 3-5, a POSITA would have been motivated to incorporate Li’s partial encryption scheme into Hagai’s multimedia playback system. EX1003, ¶174. The combination would have yielded predictable results, enabling secure playback of protected content using well-known partial encryption techniques without requiring changes to Hagai’s existing byte-range-based retrieval and playback architecture. *Id.*

Partial encryption was well-known to a POSITA before January 2007. EX1003, ¶175; *supra* II.A.1.d. Indeed, the specification of the 141 patent does not disclose any novel partial encryption or decryption technique but instead incorporates prior art by reference—including Li, which provides a detailed discussion of partial encryption. EX1003, ¶175.

G. Claim 7: “The playback device of claim 1, wherein playing back the buffered video data from the requested byte ranges provides fast forward trick play.”

Hagai discloses claim 7. EX1003, ¶176.

Hagai discloses that the playback start position table enables the playback device to request, receive, and play back data from arbitrary points in the stream. EX1004, 9:63-10:4, 10:17-21; *supra* IV.B.11. Hagai further discloses that the playback start position table can include start positions corresponding to specific times (e.g., second 1, second 2, or second 3), which the user may specify. EX1004, 9:47-54. Based on the desired time, the playback device requests and receives data from the server. EX1004, 9:63-10:7. The playback device stores the data in a buffer as it is received, then plays back the buffered data. EX1004, 10:4-10; *supra* IV.B.12; EX1003, ¶177.

A POSITA would have understood that, because Hagai enables playback from arbitrary points (for example, valid entry points to the stream that are selected by the user), the playback device could perform a series of jump playback operations forward from the present playback location and play back only buffered key frames or intra frames. EX1004, 8:4-10, EX1003, ¶178. A POSITA would have understood a series of jump playback operations displaying only these buffered key frames or intra frames (i.e., the I-frames at the start of each segment) to be a fast-forward operation because what is displayed to the user mimics the fast-forward operation in

physical media, which was a known operation. EX1004, 8:52-65, 9:50-62, 14:29-32; EX1003, ¶178; *supra* §§ II.A.1.b, II.A.1.c.

H. Claim 8: “The playback device of claim 1, wherein playing back the buffered video data from the requested byte ranges provides fast rewind trick play.”

Hagai discloses claim 8. EX1003, ¶179.

A POSITA would have understood that because Hagai enables playback from arbitrary points the playback device could play back buffered key frames or intra frames to perform a series of jump playback operations to an earlier playback location. EX1004, 8:4-10, EX1003, ¶180; *supra* § IV.G. A POSITA would have understood a series of jump playback operations displaying only these buffered key frames or intra frames (i.e., the buffered I-frames) to be a fast-rewind operation because what is displayed to the user mimics the fast-rewind operation in physical media. EX1004, 8:52-65, 9:50-62, 14:29-32; EX1003, ¶180; *supra* §§ II.A.1.b, II.A.1.c.

I. Claim 10

- 1. [10.pre], [10.a]: “The playback device of claim 1, wherein the application is further capable of causing the processor to perform the steps of: selecting a subtitle track from the multiple subtitle tracks;”**

Hagai combined with Li discloses limitations [10.pre] and [10.a]. EX1003, ¶181.

As discussed above, a POSITA would have understood the server of Hagai would store the playback-speed/file mapping table and multimedia file of Hagai, as well as the DMNU chunks of Li. EX1005, 23:3-24:10, Fig. 2.5; EX1003, ¶182; *supra* §§ IV.B.5, IV.B.8. A POSITA would have understood that Hagai’s multimedia file could store Li’s subtitle tracks, which would be synchronized with the audio and video tracks during playback. EX1003, ¶182. Li discloses that the SubtitleTrack chunk is used to identify multiple available subtitle tracks. EX1005, 23:63-24:2; *supra* § IV.B.5; EX1003, ¶183. Li further discloses that a subtitle track of the multimedia file can be chosen for display, and that the SubtitleSelectAction chunk contains the information needed to select a subtitle track. EX1005, 24:25-31, 52:58-64; EX1003, ¶184. A POSITA would have understood that the playback device would only select one subtitle track for playback at a time because it would be undesirable to display them all at once (e.g., in several different languages). EX1003, ¶185.

Moreover, a POSITA would have been motivated to store Li's DMNU chunk, which contains the SubtitleSelectAction chunk, on Hagai's server and use the SubtitleSelectAction chunk to select a subtitle track. EX1005, 23:3-24:10, Fig. 2.5; EX1003, ¶186. In the combined configuration, the server of Hagai would store its multimedia file and playback-speed/file mapping table, and also Li's DMNU chunk. EX1003, ¶186; *supra* §§ IV.B.5, IV.B.8. A POSITA would have understood that modifying the server to store the DMNU chunk of Li is a combination of prior art elements according to known methods to achieve predictable results and would involve saving data chunks, in addition to the playback-speed/file mapping table and the multimedia file of Hagai, on Hagai's server. EX1004, 1:29-45, 2:19-25; EX1005, 23:67-24:2, 24:19-31, Fig. 2.1; *supra* § IV.B.5; EX1003, ¶186. Moreover, Hagai already discloses the selection of a video track, and Li provides another mechanism, via the SubtitleSelectAction chunk, to select a subtitle track. EX1005, 24:20-24. Thus, modifying Hagai such that the server stores Li's SubtitleSelectAction chunks, and then selecting and using that information as recited in claims [10.a], [10.b], and [10.c] would be a reasonable modification that would provide expected results.

2. [10.b]: “requesting data from the selected subtitle track from the remote server system, where the data contains subtitle text; and”

Hagai combined with Li discloses limitation [10.b]. EX1003, ¶187.

As discussed above, a POSITA would have understood that the server would store Hagai's playback-speed/file mapping table and multimedia file and Li's DMNU chunks that contain the SubtitleTrack chunk identifying available subtitle tracks. EX1003, ¶188; *supra* §§ IV.B.5, IV.B.8, IV.I.1.

Li discloses that the stored subtitle tracks are encoded as a series of subtitle chunks that "are generated from text subtitles." EX1005, 29:16, 4:42-50. Each subtitle chunk includes "information capable of being represented as text on a display." EX1005, 4:45-46. Thus, a POSITA would have understood that the server of Hagai would also store data containing subtitle text. EX1003, ¶189.

A POSITA would further understand that when the playback device requests a byte range for a selected video track, the request would further include a request for the selected subtitle track because the playback device would require the necessary subtitle information to display subtitles corresponding to, and in conjunction with, the selected video content. EX1003, ¶190; *supra* §§ IV.B.9, IV.B.11.

3. [10.c]: "superimposing the subtitle text over video during playback."

Hagai combined with Li discloses limitation [10.c]. EX1003, ¶191.

Li discloses superimposing subtitle text over decoded video, then displaying the video alongside the superimposed subtitles. EX1005, 53:5-8; EX1003, ¶192. Thus, a POSITA would have understood that when the playback device of Hagai

plays the video, the playback device would utilize the requested subtitle track data from the server to superimpose the text of the subtitles over the video, implementing a conventional technique. EX1003, ¶192; *see supra* § IV.I.1.

J. Claim 20:

1. [20.pre]

To the extent the preamble of claim 20 is limiting, Hagai renders it obvious. EX1003, ¶193. Hagai discloses a method for playback on a playback device of audio and video content received from a server using HTTP. EX1004, 2:55-57, 7:64-67, 10:8-12, Figs. 2, 4; EX1003, ¶193; *supra* § IV.B.1.

2. [20.a]–[20.k] and [20.k(i)]

The method steps recited in limitations [20.a] through [20.k(i)] are substantively identical to the limitations recited in limitations [1.b(i)]–[1.b(xi)(1)], respectively. EX1003, ¶194. For the reasons described above, incorporated herein by reference, Hagai combined with Li renders these limitations of claim 20 obvious. Each of claim limitations [20.b], [20.d], [20.e], [20.h], and [20.j] recites that the step is performed “using the playback device,” which is not explicitly recited in limitations [1.b(i)]–[1.b(xi)(1)]. A POSITA would have understood that the method of claim 20 would be performed on the playback device described in the analysis of claim 1 and as described above in limitation [20.pre]. EX1003, ¶194.

3. [20.k(ii)]–[20.k(v)]

The method steps recited in limitations [20.k(ii)] through [20.k(v)] are substantively identical to the limitations recited in limitations [1.b(xi)(2)]–[1.b(xi)(5)], respectively. EX1003, ¶195. For the reasons discussed above for limitations [1.b(xi)(2)]–[1.b(xi)(5)], Hagai discloses the substantively identical limitations of claim 20. *Supra* §§ IV.J.2, IV.B.16–IV.B.19; EX1003, ¶195.

Each of claim limitations [20.k(iii)] through [20.k(v)] recites that the step is performed “using the playback device,” which is not explicitly recited in limitations [1.b(xi)(2)]–[1.b(xi)(5)]. A POSITA would have understood that the method of claim 20 would be performed on the playback device described in the analysis of claim 1 and as described above in limitation [20.pre]. EX1003, ¶195.

K. Claim 22

For the reasons discussed above for claims 3 and 20, claim 22 is unpatentable. *Supra* §§ IV.C, IV.J; EX1003, ¶196. For the reasons discussed in claim 3, a POSITA would have understood that the playback device requests the DRM header. *Id.*

L. Claim 23

For the reasons discussed above for claims 4 and 20, claim 23 is unpatentable. *Supra* §§ IV.D, IV.J; EX1003, ¶197. For the reasons discussed in claim 4, a POSITA would have understood that, once it requests and receives the DRM header, the

playback device would obtain the encryption key and decrypt the DRM header. EX1005, 52:41-55; *supra* § IV.D.

M. Claim 24

For the reasons discussed above for claims 5 and 20, claim 24 is unpatentable. *Supra* §§ IV.E, IV.J; EX1003, ¶198. For the reasons discussed in [5.b], a POSITA would have understood that the playback device of Hagai must decrypt the encoded video frames using a key prior to decoding because the playback device cannot play back the data of the video chunk in an encrypted form. EX1005, 52:49-55; *supra* § IV.E.2.

N. Claim 25

For the reasons discussed above for claims 6 and 24, claim 25 is unpatentable. *Supra* §§ IV.F, IV.M; EX1003, ¶199. Li discloses that the playback device uses DRM chunks, which reference “the decryption key that can be used to decrypt” the encrypted portions of the frame, to play encrypted video tracks. EX1005, 27:56-59, 28:14-15; *supra* § IV.F.3. For the reasons discussed above for limitation [6.c], a POSITA would have understood that the playback device uses the references in the DRM chunk to identify the encrypted portions of video frames. *Supra* § IV.F.3.

O. Claim 26

For the reasons discussed above for claims 7 and 20, claim 26 is unpatentable. *Supra* §§ IV.G, IV.J; EX1003, ¶200.

P. Claim 27

For the reasons discussed above for claims 8 and 20, claim 27 is unpatentable.

Supra §§ IV.H, IV.J; EX1003, ¶201.

Q. Claim 29

For the reasons discussed above for claims 10 and 20, claim 29 is unpatentable. *Supra* §§ IV.I, IV.J; EX1003, ¶202.

V. GROUND 2: Claims 9, 12-14, 16-18 and 28 are obvious over Hagai, Li, Park, and General Knowledge of a POSITA

A. Motivation to Combine

A POSITA would have been motivated to combine Hagai and Li for the reasons discussed in Ground 1. *Supra* §§ IV.A; EX1003, ¶203. A POSITA would have been further motivated to combine Hagai and Li with the buffer management technique taught by Park. EX1003, ¶203.

Hagai describes a client-server streaming architecture that enables seeking to arbitrary positions in a media stream using a playback start position table to determine appropriate byte ranges to request via HTTP. EX1004, 8:1-10; Claim 1. Hagai also teaches that the client buffers received stream data, controls playback based on data levels stored in the buffer levels, and monitors levels of buffered data “to avoid degradation in quality of played data due to interruptions of data.” *Id.*, 6:10-14, 6:22-31, 16:48-51; *supra* § IV.B.13. Li complements Hagai by disclosing the use of metadata—such as stream descriptors and DRM headers—to coordinate

playback and manage access. EX1005, Abstract, 1:18-22, 12:9-36, 27:42-28:15, 51:60-64, Fig. 1; EX1003, ¶204.

While neither Hagai nor Li explicitly discloses discarding previously buffered audio and video data in response to a received trick play instruction, like a seek command, Park does. EX1003, ¶205. Park discloses a “method of playout buffer management” in which the client flushes the playout buffer when a playback command, like fast-forward, fast-reverse, or a change in AV program, triggers a transition to a new stream segment. EX1006, [0012], [0026], Claim 7. Park explains that “[f]lushing the playout buffer of residual, e.g., unwanted, AV program material eliminates playout buffer delay, minimizing the reaction time of the system to user commands that change the AV stream.” *Id.*, [0026].

A POSITA would have been motivated to combine the Hagai-Li combination with Park. EX1003, ¶206. The Hagai-Li combination discloses buffer management techniques. EX1004, 6:10-14, 6:22-31, 16:48-51. The combined system would further incorporate Park’s buffer management technique of flushing the audio and video data stored in the buffer. EX1003, ¶206. Since Hagai already discloses buffer management techniques, it would have been straightforward to add the additional buffer management technique taught by Park. *Id.* This additional technique would minimize playout buffer delays and improve responsiveness of the playback device of the Hagai-Li combination. *Id.*

A POSITA would have had a reasonable expectation of success combining Hagai and Li with Park because Park uses a well-known technique—flushing from the buffer previously requested audio and video data that is no longer needed—to manage the efficiency of the playback device’s buffer. EX1003, ¶207. The combination of these teachings would have yielded predictable results because it involves conventional and well-known buffer management techniques in playback systems. *Id.*; EX1017, 13:49-61, Fig. 5.

B. Claims 9 and 28⁴

- 1. Claim 9: “The playback device of claim 1, wherein the application is further capable of causing the processor to perform the step of discarding audio and video data contained within the buffer in response to a received seek instruction.”**

Hagai in combination with Li and Park renders obvious claims 9 and 28. EX1003, ¶208.

As discussed in limitations [1.b(x)], Hagai discloses that the playback device buffers received audio and video data, and a “reception buffer monitor” monitors the quantity of stored audio and video data in the buffer. EX1004, 6:10-14, 16:48-51; EX1003, ¶209; *supra* § IV.B.13. When the reception buffer monitor detects that the

⁴ Claim 28, which recites elements that overlap with those of claim 9, is provided in the Appendix.

quantity of buffered data exceeds a predetermined value, the playback device restarts playback. EX1004, 17:1-7. To the extent Hagai does not explicitly disclose discarding buffered data in response to a seek instruction, a POSITA would look to Park for its buffer management techniques to minimize latency and avoid playback of outdated data. EX1003, ¶209; EX1017, 13:49-61, Fig. 5.

Park discloses a “method of playout buffer management” that “flush[es] the playout buffer.” EX1006, [0012]. A POSITA would have understood that “flushing the playout buffer” of Park is equivalent to “discarding the audio and video data contained within the buffer” because both result in clearing the playback device buffer of outdated data. EX1003, ¶210.

Park further discloses that the buffer is flushed in response to a group of commands, such as a fast-forward or rewind command. EX1006, [0012]. A POSITA would have understood that Park’s flushing technique applies to a received seek instruction because a seek instruction similarly leaves buffered data outdated by jumping to a new playback location. EX1006, [0026]; EX1003, ¶211.

As discussed above, a POSITA would have been motivated to combine the Hagai-Li system with Park because Hagai already discloses buffer management techniques, and incorporating additional buffer management techniques disclosed in Park would have been a straightforward modification to minimize playout buffer

delays and improve responsiveness of the playback device of the Hagai-Li combination. EX1003, ¶212; EX1006, [0023], [0027]; *supra* § V.A.

C. Claim 12

1. [12.pre]

For the reasons discussed above for limitation [1.pre], limitation [12.pre] is unpatentable. *Supra* § IV.B.1; EX1003, ¶213.

2. [12.a]

For the reasons discussed above for limitation [1.a], limitation [12.a] is unpatentable. *Supra* § IV.B.2; EX1003, ¶214.

3. [12.b]

For the reasons discussed above for limitation [1.b], limitation [12.b] is unpatentable. *Supra* § IV.B.3; EX1003, ¶215.

4. [12.b(i)]

For the reasons discussed above for limitation [1.b(i)], limitation [12.b(i)] is unpatentable. *Supra* § IV.B.4; EX1003, ¶216.

5. [12.b(ii)]

For the reasons discussed above for limitation [1.b(ii)], limitation [12.b(ii)] is unpatentable. *Supra* § IV.B.5; EX1003, ¶217.

6. [12.b(iii)]

For the reasons discussed above for limitation [1.b(iii)], limitation [12.b(iii)] is unpatentable. *Supra* § IV.B.6; EX1003, ¶218.

7. [12.b(iv)]

For the reasons discussed above for limitation [1.b(iv)], Hagai in combination with Li renders obvious requesting a header describing the selected video track. *Supra* § IV.B.7; EX1004, 8:11-18; EX1003, ¶219.

Moreover, for the reasons discussed above for claim 3, Hagai in combination with Li renders obvious that the requested header comprises a DRM header. *Supra* § IV.C; EX1003, ¶¶220-21.

8. [12.b(v)]

For the reasons discussed above for claim 4, limitation [12.b(v)] is unpatentable. *Supra* § IV.D; EX1003, ¶222.

9. [12.b(vi)]

For the reasons discussed above for limitation [1.b(v)], limitation [12.b(vi)] is unpatentable. *Supra* § IV.B.8; EX1003, ¶223.

10. [12.b(vii)]

For the reasons discussed above for limitation [1.b(vi)], limitation [12.b(vii)] is unpatentable. *Supra* § IV.B.9; EX1003, ¶224.

11. [12.b(viii)]

For the reasons discussed above for limitation [1.b(vii)], limitation [12.b(viii)] is unpatentable. *Supra* § IV.B.10; EX1003, ¶225.

12. [12.b(ix)]

Hagai discloses limitation [12.b(ix)]. EX1003, ¶226.

As discussed above, after sending byte range requests to the server, the playback device stores the received data in a buffer. EX1004, 10:4-7; *supra* § IV.B.12. Hagai discloses that the reception means of the playback device “receives a transmission response” that includes the “stream data,” which it stores “in the reception buffer” before playing stream data. EX1004, 10:4-10, Fig. 2. A POSITA would have understood that, by receiving data and storing it for playback, Hagai’s system is “creating a buffer.” EX1003, ¶227.

13. [12.b(x)]

For the reasons discussed above for limitation [1.b(viii)], limitation [12.b(x)] is unpatentable. *Supra* § IV.B.11; EX1003, ¶228.

14. [12.b(xi)]

For the reasons discussed above for limitation [1.b(ix)], limitation [12.b(xi)] is unpatentable. *Supra* § IV.B.12; EX1003, ¶229.

15. [12.b(xii)]

For the reasons discussed above for limitation [1.b(x)], limitation [12.b(xii)] is unpatentable. *Supra* § IV.B.13; EX1003, ¶230.

16. [12.b(xiii)]

For the reasons discussed above for claim 5, limitation [12.b(xiii)] is unpatentable. *Supra* § IV.E.2; EX1003, ¶231.

17. [12.b(xiv)]

For the reasons discussed above for limitation [1.b(x)] and claim 5, limitation [12.b(xiv)] is unpatentable. *Supra* §§ IV.B.13, IV.E; EX1003, ¶232.

18. [12.b(xv)]

For the reasons discussed above for limitation [1.b(xi)], limitation [12.b(xv)] is unpatentable. *Supra* § IV.B.14; EX1003, ¶233.

19. [12.b(xv)(1)]

For the reasons discussed above for limitation [1.b(xi)(1)], limitation [12.b(xv)(1)] is unpatentable. *Supra* § IV.B.15; EX1003, ¶234.

20. [12.b(xv)(2)]

For the reasons discussed above for claim 9, limitation [12.b(xv)(2)] is unpatentable. *Supra* § V.B; EX1003, ¶235.

21. [12.b(xv)(3)]–[12.b(xv)(8)]

Upon receiving a new seek instruction, Hagai’s playback device repeats the same sequence of operations used for playback before the new instruction. It determines and requests new byte ranges corresponding to the new playback location, buffers the data, decrypts the encrypted video frames, and, once sufficient data is buffered, resumes playback from the new playback location. EX1004, 9:4-10:21, 11:5-35; *supra* § IV.E. These operations correspond to limitations [12.b(viii)] and [12.b(x)]–[12.b(xiv)], except that they are performed in relation to the “new playback location” selected by the client device. Accordingly, these elements are

obvious for the same reasons stated in [12.b(viii)] and [12.b(x)]-[12.b(xiv)]. *Supra* §§ V.C.11, V.C.13-V.C.17; EX1003, ¶236.

D. Claim 13

For the reasons discussed above for limitation [12.b(xiii)] and claim 5, claim 13 is unpatentable. *Supra* §§ V.C.11, IV.E; EX1003, ¶237.

E. Claim 14

For the reasons discussed above for claim 6, claim 14 is unpatentable. *Supra* §§V.C.11 IV.F, V.C.11; EX1003, ¶238.

F. Claim 16

For the reasons discussed above for claim 7, claim 16 is unpatentable. *Supra* §§V.C.11 IV.G, V.C.11; EX1003, ¶239.

G. Claim 17

For the reasons discussed above for claim 8, claim 17 is unpatentable. *Supra* §§V.C.11 IV.H, V.C.11; EX1003, ¶240.

H. Claim 18

For the reasons discussed above for claim 10, claim 18 is unpatentable. *Supra* §§V.C.11 IV.I, V.C.11; EX1003, ¶241.

VI. GROUND 3: CLAIMS 2, 11, 21, AND 30 ARE OBVIOUS OVER HAGAI, LI, SCHMITZ, AND GENERAL KNOWLEDGE OF A POSITA

A. Motivation to Combine

A POSITA would have been motivated to combine Hagai and Li for the reasons discussed in Ground 1. *Supra* § IV.A; EX1003, ¶242. A POSITA would have been further motivated to enhance the Hagai-Li combination by incorporating the server request management techniques taught by Schmitz. *Id.*

Hagai describes a client-server streaming architecture that enables seeking (i.e., navigating to) arbitrary positions in a media stream using a playback start position table to determine appropriate byte ranges to request via HTTP. EX1004, 8:1-10; *supra* §§ IV.B.9-IV.B.11. While neither Hagai nor Li explicitly describes how to handle existing byte range requests rendered obsolete by a new seek instruction or use asynchronous requests to the server for byte ranges from the selected video track and audio track, Schmitz does. EX1003, ¶243.

Schmitz is directed to navigation through multimedia content including time-linear media, such as video and audio, and non-linear media, like encyclopedias. EX1007, 1:25-30, 1:45-52, 6:18-22. Prior systems typically restricted users to linear, sequential navigation. *Id.*, 2:8-14. Schmitz's system improves upon the prior art by using cancelable asynchronous requests to allow arbitrary navigation and rapid content retrieval based on user interaction. *Id.*, 1:25-30. Schmitz explains that

“the present invention does not bind retrieval of the video context pictures synchronously to the user navigation,” thereby enabling fast system response to seek operations. *Id.*, 13:30-32. These teachings naturally align with the Hagai-Li combination, which supports media playback from arbitrary points. EX1003, ¶¶244-45.

Though Schmitz’s techniques were novel to non-linear media (like encyclopedias), it states that the techniques apply equally to time-linear media (like video). EX1007, 1:67-2:7, 6:1-10. Traditionally, time-linear content was constrained by physical media like tapes, requiring sequential, synchronous operation. EX1007, 1:67-2:7. Schmitz improves upon this by enabling random access to time-linear media through asynchronous requests, allowing users to seek freely without waiting for prior requests to be fulfilled. EX1007, 6:15-23. This is functionally analogous to the jump playback operation supported by the Hagai-Li combination. EX1003, ¶246.

Schmitz discloses that the multimedia system includes a client device capable of canceling, or flushing, obsolete requests previously sent to the server in response to user actions. EX1007, 1:9-14, 6:58-65, 7:60-64, Abstract; *supra* § III.D; EX1003, ¶247. Schmitz also discloses that the client sends asynchronous requests to the server. EX1007, 6:15-65, 7:60-64, 21:56-59, 23:45-57, 4:60-67, 8:48-64, Claim 1; *supra* § III.D; EX1003, ¶247.

A POSITA would have been motivated to combine the Hagai-Li combination with Schmitz's asynchronous request and request-flushing techniques. EX1003, ¶248. Hagai already discloses sending HTTP requests for byte ranges of selected media tracks, and a POSITA would have known that retaining stale or obsolete requests makes no technical sense and thus they would need to be discarded. EX1003, ¶248. To the extent a POSITA needed explicit teachings of such a technique, Schmitz expressly discloses such behavior, and a POSITA would be motivated to implement with Hagai and Li to provide for this functionality and optimize server response time and communications between the server and playback device. *Id.*

A POSITA would have also been motivated to incorporate Schmitz's disclosure of asynchronous requests into the Hagai-Li combination. EX1003, ¶249. Because Hagai already sends HTTP requests from the playback device to the server, it would have required a simple modification to the Hagai-Li combination, which already includes a control means for controlling transmission requests, to include HTTP requests that can handle this type of operation. EX1004, 9:17-19, 9:63-67; EX1003, ¶249. For example, HTTP is extensible and request-header field names can be extended to add functionality such as flushing pending messages. This was well known and specified in HTTP 1.1 standard of 1999 (RFC 2616). EX1030, EX1003, ¶249. A POSITA would have understood that such a combination would

improve the playback functionality of the Hagai-Li combination by decreasing server response time, reducing the number of outstanding server requests, and enabling prioritization of outstanding requests by the server. EX1007, 4:60-67, 6:62-7:5, 23:45-57; EX1003, ¶249.

A POSITA would have had a reasonable expectation of success implementing both modifications. Request flushing and asynchronous requests were known solutions in the field, and their integration would have predictably improved the performance and responsiveness of the Hagai-Li system. EX1003, ¶250.

B. Claims 11 and 30⁵

- 1. Claim 11: “The playback device of claim 1, wherein the application is further capable of causing the processor to perform the step of flushing previous byte range requests in response to the received seek instruction.”**

Hagai in combination with Li and Schmitz renders obvious claims 11 and 30. EX1003, ¶251. As explained above, Hagai in combination with Li discloses that the playback device requests from the server byte ranges from the selected video and audio tracks. *Supra* §§ IV.B.11, IV.J.

⁵ Claim 30, which recites elements that overlap with those of claim 11, is provided in the Appendix.

To the extent that Hagai and Li do not disclose flushing previous byte range requests in response to a received seek instruction, Schmitz does. EX1003, ¶252. Schmitz discloses a “media server request mechanism” that “permits cancellation of outstanding queued requests that have become obsolete as the user navigates out of the area for which the request was needed.” EX1007, 1:9-14; Abstract. A POSITA would have understood that cancelling obsolete requests to be “flushing” them because it discards requests that are lower priority than new requests. EX1003, ¶252. Schmitz further discloses that the client cancels obsolete requests related to subsets of data, such as subsets of bytes representing portions of images. EX1007, Abstract, 12:29-49. Accordingly, a POSITA would have understood that Schmitz’s request cancellation would apply to the byte range requests of Hagai-Li. EX1003, ¶252.

As discussed above, a POSITA would have been motivated to combine Schmitz with the Hagai-Li combination because a POSITA would know that obsolete requests would need to be discarded to maintain optimal server response times. EX1003, ¶253. To the extent this would not be understood through Hagai or Li, a POSITA would have looked to Schmitz for an explicit teaching and would have had a reasonable expectation of success because it would require only a simple addition to the Hagai-Li combination of a known solution. *Id.*; *supra* § VI.A.

C. Claims 2 and 21⁶

Hagai in view of Li and Schmitz renders claims 2 and 21 obvious. EX1003, ¶254. As discussed above, Hagai in view of Li teaches every limitation of claims 1 and 20. *Supra* §§ IV.B, IV.J.

- 1. Claim 2: “The playback device of claim 1, wherein the application is further capable of causing the processor to asynchronously request byte ranges from the selected video track and the selected audio track from the remote server.”**

Hagai in combination with Li and Schmitz renders obvious claims 2 and 21. EX1003, ¶255.

As explained above, Hagai in combination with Li renders obvious that the playback device requests byte ranges from the selected video track and selected audio track from the remote server. *Supra* §§ IV.B.11, IV.J; EX1003, ¶256.

At the time of the invention, requests could be sent synchronously or asynchronously. EX1003, ¶257. Thus, a POSITA would have understood that the Hagai-Li combination teaches these limitations based on the knowledge of a skilled artisan, even without explicit disclosure about the type of request. *Id.* But even if asynchronously requesting byte ranges is not expressly disclosed by Hagai or Li, it

⁶ Claim 21, which recites elements that overlap with those of claim 2, is provided in the Appendix.

is in Schmitz, which a POSITA would have been motivated to look to for detail about incorporating this standard functionality. *Id.*

A POSITA would have understood that Schmitz discloses asynchronous requests because it enables sending new requests independent of the status of prior requests. EX1007, 21:56-57; EX1003, ¶258; EX1001, 2:31-34.

Schmitz discloses a multimedia system including a client module that sends asynchronous requests to the multimedia server. EX1007, 6:15-65, 21:56-59, 8:48-64. A POSITA would have understood that the client module generates asynchronous requests because the client module can generate and transmit requests to the multimedia server independent of outstanding requests. EX1007, 6:23-31, 21:56-59; EX1003, ¶259. A POSITA would further understand that the requests transmitted by the client module are asynchronous requests because the client module can prioritize, re-prioritize, and cancel requests pending at the multimedia server. EX1007, 21:56-57; 23:45-57; EX1003, ¶259.

As discussed in Section VI.A, a POSITA would have been motivated to combine Schmitz with the Hagai-Li system because Hagai already discloses sending HTTP requests for byte ranges of selected video and audio tracks, and sending asynchronous requests to the server was a well-known and conventional technique. EX1003, ¶260; *supra* § VI.A. A POSITA would have further understood that such a combination would have been straightforward and would have had a reasonable

expectation of success because it would require only a simple modification to the Hagai-Li combination, which already includes a control means for controlling transmission requests. EX1004, 9:17-19, 63-67; EX1003, ¶260; *supra* § VI.A.

VII. GROUND 4: Claims 15 and 19 are obvious over Hagai, Li, Park, Schmitz, and General Knowledge of a POSITA

A. Motivation to Combine

A POSITA would have been motivated to combine Hagai, Li, and Park for the reasons stated in Grounds 1 and 2. *Supra* §§ IV.A, V.A. A POSITA also would have been motivated to combine Schmitz with Hagai and Li as explained above in Ground 3. *Supra* § VI.A. A POSITA would have further been motivated to combine Hagai, Li, and Park with Schmitz. EX1003, ¶261. A POSITA would be motivated to combine both Park and Schmitz with the Hagai-Li combination because they introduce independent techniques to reduce delays in different aspects of the data transmission process. While Schmitz teaches techniques for improving response time to requests being sent out by the client device (i.e., it makes efficiency improvements in communications with the server), Park offers an additional technique to improve efficiency of received data by minimizing playout buffer delays and improving responsiveness of the playback device (i.e., improvements to buffer management and playback device performance). *Supra* §§ IV.A, V.A, VI.A; EX1003, ¶261. As a result, the implementation of both techniques would improve the overall performance of the Hagai-Li combination. EX1003, ¶261.

B. Claim 15

The combination of Schmitz in view of Hagai, Li, Park, and Schmitz renders claim 15 obvious. EX1003, ¶262. As discussed above, Hagai in view of Li and Park teaches every limitation of claim 12. *Supra* § V.C.

For the reasons discussed above for claims 2 and 21, Hagai in combination with Li, Park, and Schmitz renders obvious claim 15. *Supra* § VI.C; EX1003, ¶263. As noted above, the inclusion of Park in the Hagai-Li combination does not impact the combination of Schmitz with Hagai and Li because it teaches how to efficiently handle received data and has no effect on Schmitz's techniques for improving requests made by the client. *Supra* § VII.A.

C. Claim 19

The combination of Schmitz in view of Hagai, Li, Park, and Schmitz renders claim 19 obvious. EX1003, ¶264. As discussed above, Hagai in view of Li and Park teaches every limitation of claim 12. *Supra* § V.C.

For the reasons discussed above for claims 11 and 30, Hagai in combination with Li, Park, and Schmitz renders obvious claim 19. *Supra* § VI.B; EX1003, ¶265. As noted above, the inclusion of Park in the Hagai-Li combination has no impact on the combination of Schmitz with Hagai and Li. *Supra* § VII.B.

VIII. CONCLUSION

For these reasons, Petitioner requests *inter partes* review and cancellation of the challenged claims. EX1003, ¶266.

IX. MANDATORY NOTICES AND FEES

A. Real Party-in-Interest

Petitioner is the real party-in-interest. No other party directed, controlled, or funded this IPR proceeding.

B. Related Matters

Petitioner is aware of the following additional related matters involving the 141 Patent and/or related patents:

Case Caption	Forum	Patents
<i>DivX, LLC v. Amazon.com, Inc.</i> , 1:24-cv-02061-CMH-LRV	Eastern District of Virginia	U.S. Patent Nos. 10,412,141; 10,715,806; 9,955,195; 11,611,785; 10,542,303; 11,245,938; and 12,184,943

C. Lead and Backup Counsel

Pursuant to 37 C.F.R. § 42.8(b)(3) and 42.10(a), Petitioner designates J. David Hadden, Reg. No. 40,629 as lead counsel and Saina Shamilov, Reg. No. 48,266, Allen Wang, Reg. No. 68,456, as backup counsel, each of Fenwick & West LLP.

D. Service Information

Petitioner consents to service by electronic mail at:

DivX-IPR@fenwick.com.

Petitioner's counsel may also be served by mail or hand delivery at Fenwick & West LLP, 801 California St, Mountain View, CA 94041. Petitioner's counsel may be reached by telephone at (650) 988-8500.

E. Fees

The Office is authorized to charge fees for this Petition to Deposit Account 19-2555.

Dated: June 30, 2025

FENWICK & WEST LLP

/J. David Hadden/

J. David Hadden
Reg. No. 40,629

Attorneys for Petitioner
Amazon.com, Inc. and
Amazon Web Services, Inc.

CERTIFICATION OF WORD COUNT

The undersigned certifies pursuant to 37 C.F.R. § 42.24 that the foregoing Petition for *Inter Partes* Review, excluding any table of contents, mandatory notices under 37 C.F.R. § 42.8, certificates of service or word count, or appendix of exhibits, contains 13,931 words according to the word-processing program used to prepare this document (Microsoft Word).

Dated: June 30, 2025

FENWICK & WEST LLP

/J. David Hadden/
J. David Hadden
Reg. No. 40,629

Attorneys for Petitioner
Amazon.com, Inc. and
Amazon Web Services, Inc.

**CERTIFICATE OF SERVICE ON PATENT OWNER
UNDER 37 C.F.R. § 42.105**

I hereby certify, pursuant to 37 C.F.R. Sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 10,412,141**, including all exhibits (**Nos. 1001-1035**), is being served via Federal Express on June 30, 2025, upon Patent Owner by serving the correspondence address of record with the USPTO as follows:

71897 - KPPB LLP
3780 Kilroy Airport Way, Suite 320
Long Beach, CA 90806
UNITED STATES

The foregoing was also served via Federal Express upon counsel of record for Patent Owner in the litigation pending before the U.S. District Court for the Eastern District of Virginia entitled *DivX, LLC v. Amazon.com, Inc., and Amazon Web Services, Inc.*, No. 1:24-cv-020610CMH-LRV as follows:

Charles B. Molster, III
THE LAW OFFICES OF CHARLES B. MOLSTER, III PLLC
2141 Wisconsin Avenue, N.W., Suite M
Washington, D.C. 20007

Dated: June 30, 2025

FENWICK & WEST LLP

/J. David Hadden/

J. David Hadden

Reg. No. 40,629

Attorneys for Petitioner
Amazon.com, Inc. and
Amazon Web Services, Inc.

APPENDIX: LIST OF CHALLENGED CLAIMS

141 patent, Claim 1	
[1.pre]	A playback device, comprising:
[1.a]	a processor; and
[1.b]	a non-volatile storage containing an application for causing the processor to perform the steps of:
[1.b(i)]	establishing at least one connection for communicating with a remote server system;
[1.b(ii)]	obtaining information from a remote server system describing at least one video track, multiple audio tracks, and multiple subtitle tracks;
[1.b(iii)]	selecting a video track from the at least one video track;
[1.b(iv)]	requesting a header describing the selected video track;
[1.b(v)]	selecting an audio track from the multiple audio tracks;
[1.b(vi)]	obtaining index information indicating the locations of audio and video data within the selected audio and video tracks;
[1.b(vii)]	determining byte ranges to request from the selected audio and video tracks using the index information;
[1.b(viii)]	requesting byte ranges from the selected video track and the selected audio track from the remote server system;
[1.b(ix)]	buffering received bytes of information comprising audio and video data;

[1.b(x)]	checking that sufficient data is buffered to commence playback and playing back the buffered audio and video data;
[1.b(xi)]	responding to a received seek instruction by:
[1.b(xi)(1)]	pausing playback;
[1.b(xi)(2)]	determining byte ranges to request from the selected audio and video tracks based upon a new playback location using the index information;
[1.b(xi)(3)]	requesting byte ranges required to play the selected audio and video tracks from the new playback location from the remote server;
[1.b(xi)(4)]	buffering received bytes of information comprising audio and video data pending resumption of playback; and
[1.b(xi)(5)]	checking that sufficient data is buffered to commence playback and playing back the buffered audio and video data.
141 patent, Claim 2	
2	The playback device of claim 1, wherein the application is further capable of causing the processor to asynchronously request byte ranges from the selected video track and the selected audio track from the remote server.
141 patent, Claim 3	
3	The playback device of claim 1, wherein the step of requesting a header describing the selected video track comprises the step of requesting a DRM header.
141 patent, Claim 4	
4	The playback device of claim 3, wherein the application is further capable of causing the processor to perform the step of decrypting the DRM header.

141 patent, Claim 5	
[5.pre]	The playback device of claim 3, wherein:
[5.a]	the DRM header identifies at least one key; and
[5.b]	the application is further capable of causing the processor to perform the steps of decrypting encrypted frames of video using the at least one key prior to decoding.
141 patent, Claim 6	
[6.pre]	The playback device of claim 5, wherein:
[6.a]	encrypted frames of video are only partially encrypted;
[6.b]	the received bytes of information contain references to portions of video frames that are encrypted; and
[6.c]	the step of decrypting encrypted frames of video using the at least one key prior to decoding further comprises decrypting encrypted portions of frames of video using the references to the portions of the video frames that are encrypted.
141 patent, Claim 7	
7	The playback device of claim 1, wherein playing back the buffered video data from the requested byte ranges provides fast forward trick play.
141 patent, Claim 8	
8	The playback device of claim 1, wherein playing back the buffered video data from the requested byte ranges provides fast rewind trick play.

141 patent, Claim 9	
9	The playback device of claim 1, wherein the application is further capable of causing the processor to perform the step of discarding audio and video data contained within the buffer in response to a received seek instruction.
141 patent, Claim 10	
[10.pre]	The playback device of claim 1, wherein the application is further capable of causing the processor to perform the steps of:
[10.a]	selecting a subtitle track from the multiple subtitle tracks;
[10.b]	requesting data from the selected subtitle track from the remote server system, where the data contains subtitle text; and
[10.c]	superimposing the subtitle text over video during playback.
141 patent, Claim 11	
11	The playback device of claim 1, wherein the application is further capable of causing the processor to perform the step of flushing previous byte range requests in response to the received seek instruction.
141 patent, Claim 12	
[12.pre]	A playback device, comprising:
[12.a]	a processor; and
[12.b]	a non-volatile storage containing an application for causing the processor to perform the steps of:
[12.b(i)]	establishing at least one connection for communicating with a remote server system;

[12.b(ii)]	obtaining information from a remote server system describing at least one video track, and at least one audio track;
[12.b(iii)]	selecting a video track from the at least one video track;
[12.b(iv)]	requesting a header describing the selected video track, where the requested header comprises a DRM header;
[12.b(v)]	decrypting the DRM header;
[12.b(vi)]	selecting an audio track from the at least one audio track;
[12.b(vii)]	obtaining index information indicating the locations of audio and video data within the selected audio and video tracks;
[12.b(viii)]	determining byte ranges to request from the selected audio and video tracks using the index information;
[12.b(ix)]	creating a buffer;
[12.b(x)]	requesting byte ranges from the video track and the audio track from the remote server system;
[12.b(xi)]	buffering received bytes of information comprising audio and video data
[12.b(xii)]	checking that sufficient data is buffered to commence playback;
[12.b(xiii)]	decrypting encrypted frames of video using information from the decrypted DRM header;
[12.b(xiv)]	playing back the buffered audio and the decrypted video data;
[12.b(xv)]	responding to a received seek instruction by:
[12.b(xv)(1)]	pausing playback;

[12.b(xv)(2)]	discarding buffered audio and video data;
[12.b(xv)(3)]	determining byte ranges to request from the selected audio and video tracks based upon a new playback location using the index information;
[12.b(xv)(4)]	requesting byte ranges required to play the selected audio and video tracks from the new playback location from the remote server;
[12.b(xv)(5)]	buffering received bytes of information comprising audio and video data pending resumption of playback;
[12.b(xv)(6)]	checking that sufficient data is buffered to commence playback;
[12.b(xv)(7)]	decrypting encrypted frames of video using information from the decrypted DRM header; and
[12.b(xv)(8)]	playing back the buffered audio and decrypted video data.
141 patent, Claim 13	
[13.pre]	The playback device of claim 12, wherein:
[13.a]	the DRM header identifies at least one key; and
[13.b]	the step of decrypting encrypted frames of video using information from the decrypted DRM header comprises decrypting encrypted frames of video using the at least one key prior to decoding.
141 patent, Claim 14	
[14.pre]	The playback device of claim 13, wherein
[14.a]	encrypted frames of video are only partially encrypted;
[14.b]	the received bytes of information contain references to portions of video frames that are encrypted; and

[14.c]	the step of decrypting encrypted frames of video using the at least one key prior to decoding further comprises decrypting encrypted portions of frames of video using the references to the portions of the video frames that are encrypted.
141 patent, Claim 15	
15	The playback device of claim 12, wherein the application is further capable of causing the processor to asynchronously request byte ranges from the selected video track and the selected audio track from the remote server.
141 patent, Claim 16	
16	The playback device of claim 12, wherein playing back the buffered video data from the requested byte ranges provides fast forward trick play
141 patent, Claim 17	
17	The playback device of claim 12, wherein playing back the buffered video data from the requested byte ranges provides fast rewind trick play.
141 patent, Claim 18	
[18.pre]	The playback device of claim 12, wherein the application is further capable of causing the processor to perform the steps of:
[18.a]	obtaining information from a remote server system describing at least one subtitle track;
[18.b]	selecting a subtitle track from the at least one subtitle track;
[18.c]	requesting data from the selected subtitle track from the remote server system, where the data contains subtitle text; and

[18.d]	superimposing the subtitle text over video during playback.
141 patent, Claim 19	
19	The playback device of claim 12, wherein the application is further capable of causing the processor to perform the step of flushing previous byte range requests in response to the received seek instruction.
141 patent, Claim 20	
[20.pre]	A method of playing back content on a playback device, comprising:
[20.a]	establishing at least one connection for communicating with a remote server system using a playback device;
[20.b]	obtaining information from a remote server system using the playback device, where the obtained information describes at least one video track, multiple audio tracks, and multiple subtitle tracks;
[20.c]	selecting a video track from the at least one video track;
[20.d]	requesting a header describing the at least one video track using the playback device;
[20.e]	selecting an audio track from the multiple audio tracks using the playback device;
[20.f]	obtaining index information indicating the locations of audio and video data within the selected audio and video tracks;
[20.g]	determining byte ranges to request from the selected audio and video tracks using the index information;
[20.h]	requesting byte ranges from the selected video track and the selected audio track from the remote server system using the playback device;

[20.i]	buffering received bytes of information comprising audio and video data on the playback device;
[20.j]	checking that sufficient data is buffered to commence playback and playing back the buffered audio and video data using the playback device;
[20.k]	responding to receipt of a seek instruction at the playback device by:
[20.k(i)]	pausing playback on the playback device;
[20.k(ii)]	determining byte ranges to request from the selected audio and video tracks based upon a new playback location using the index information;
[20.k(iii)]	requesting byte ranges required to play the selected audio and video tracks from the new playback location from the remote server using the playback device;
[20.k(iv)]	buffering received bytes of information comprising audio and video data pending resumption of playback using the playback device; and
[20.k(v)]	checking that sufficient data is buffered to commence playback and playing back the buffered audio and video data using the playback device.
141 patent, Claim 21	
21	The method of claim 20, further comprising asynchronously requesting byte ranges from the selected video track and the selected audio track from the remote server using the playback device.
141 patent, Claim 22	
22	The method of claim 20, wherein requesting a header describing the selected video track further comprises requesting a DRM header using the playback device.

141 patent, Claim 23	
23	The method of claim 22, further comprising decrypting the DRM header.
141 patent, Claim 24	
[24.pre]	The method of claim 23, wherein:
[24.a]	the DRM header identifies at least one key; and
[24.b]	the method further comprises decrypting encrypted frames of video using the at least one key prior to decoding using the playback device.
141 patent, Claim 25	
[25.pre]	The method of claim 24, wherein:
[25.a]	encrypted frames of video are only partially encrypted;
[25.b]	the received bytes of information contain references to portions of video frames that are encrypted; and
[25.c]	decrypting encrypted frames of video using the at least one key prior to decoding further comprises decrypting encrypted portions of frames of video using the references to the portions of the video frames that are encrypted using the playback device.
141 patent, Claim 26	
26	The method of claim 20, wherein playing back the buffered video data from the requested byte ranges provides fast forward trick play.
141 patent, Claim 27	
27	The method of claim 20, wherein playing back the buffered video data from the requested byte ranges provides fast rewind trick play.

141 patent, Claim 28	
28	The method of claim 20, further comprising discarding audio and video data contained within the buffer in response to a received seek instruction.
141 patent, Claim 29	
[29.pre]	The method of claim 20, further comprising:
[29.a]	selecting a subtitle track from the multiple subtitle tracks using the playback device;
[29.b]	requesting data from the selected subtitle track from the remote server system using the playback device, where the data contains subtitle text; and
[29.c]	superimposing the subtitle text over video during playback using the playback device.
141 patent, Claim 30	
30	The method of claim 20, further comprising flushing previous byte range requests in response to the received seek instruction.