

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: James M. Barton, et al.
U.S. Patent No.: 6,233,389 Attorney Docket No.: 39843-0037IP1
Issue Date: May 15, 2001
Appl. Serial No.: 09/126,071
Filing Date: July 30, 1998
Title: Multimedia Time Warping System

Mail Stop Patent Board

Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

**PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES PATENT
NO. 6,233,389 PURSUANT TO 35 U.S.C. §§ 311–319, 37 C.F.R. § 42**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	REQUIREMENTS FOR IPR	5
	A. Grounds for Standing.....	5
	B. Challenge and Relief Requested.....	6
III.	BACKGROUND	8
	A. Overview of the '389 Patent	8
	B. Prior Art	11
	C. Claim Construction under 37 C.F.R. §§ 42.104(b)(3).....	13
IV.	APPLICATION OF PRIOR ART TO CHALLENGED CLAIMS	18
	A. GROUND 1: Claims 31 and 61 are Anticipated under § 102(b) by Sampat (client-side)	19
	B. GROUND 2: Claims 31 and 61 are Obvious under § 103 over Sampat (client-side) in view of VfW and SoundBlaster	51
	C. GROUND 3: Claims 31 and 61 are Anticipated under § 102(b) by Sampat (server-side)	56
	D. GROUND 4: Claims 31 and 61 are Obvious under § 103 over Sampat (server-side) in view of VfW, SoundBlaster, and Gerber	86
V.	Secondary Considerations	92
VI.	Redundancy	93
VII.	CONCLUSION.....	93
VIII.	MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1).....	95
	A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1).....	95
	B. Related Matters Under 37 C.F.R. § 42.8(b)(2)	95
	C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3).....	96
	D. Service Information	96

EXHIBITS

SE1001	U.S. Pat. No. 6,233,389 to Barton et al. (“the ’389 patent”)
SE1002	Prosecution History of the ’389 patent (Serial No. 09/126,071)
SE1003	Declaration of Declaration of John M. Strawn, Ph.D.
SE1004	U.S. Patent No. 5,557,724 to Sampat et al. (“Sampat”)
SE1005	U.S. Patent No. 5,710,895 to Gerber et al. (“Gerber”)
SE1006	Sound Blaster Pro User Reference Manual (1991) (“SoundBlaster”)
SE1007	Programmer’s Guide, Microsoft Video for Windows Development Kit (February 1993) (“VfW”)
SE1008	U.S. Patent No. 5,546,103 to Rhodes et al. (“Rhodes”)
SE1009	Definition of “Capture”, Concise Oxford Dictionary of Current English (1990)
SE1010	Definition of “Capture (of data)”, Webster's New World Dictionary of Computer Terms (1988)
SE1011	Claim Construction Order, <i>TiVo Inc. v. Echostar Communications Corp., et al.</i> , 2:04-cv-00001 (8/18/2005)
SE1012	Claim Construction Order, <i>TiVo, Inc. v. AT&T Inc., et al.</i> , 2:09- cv-00259 (10/13/2011)

- SE1013 Claim Construction Order, *TiVo, Inc. v. Verizon Comm’n, Inc. et al.*, 2:09-cv-00257 (3/12/2012)
- SE1014 Memorandum Opinion and Order, *Motorola Mobility, Inc. et al. v. TiVo, Inc.*, 5:11-cv-00053 (12/06/2012)
- SE1015 Exhibit A, Preliminary Infringement Claim Chart for U.S. Pat. No. 6,233,389, Samsung Mobile Devices (“Infringement Contentions”)
- SE1016 Prosecution History of *Ex Parte* Reexamination of claims 1, 3-5, 15-18, 20-25, 32, 34-36, 46-49, and 51-55 of the ’389 patent (Serial No. 90/007750) (“First Reexam”)
- SE1017 Prosecution History of *Ex Parte* Reexamination of claims 31 and 61 of the ’389 patent (Serial No. 90/009329) (“Second Reexam”)
- SE1018 Bescos, Jesus et al., *From Multimedia Stream Models to GUI Generation* (1997) (“Bescos”)
- SE1019 Screen capture of Amazon.com listing for Sound Blaster Pro User Reference Manual (accessed July 8, 2016)
- SE1020 Musser, John, *A Multimedia Class Library for Windows*, Dr. Dobb’s Journal (July 1993) (“Musser”)
- SE1021 Adams, Eric J., *High Noon: Big Players Ready for Video Showdown*, MacWEEK (Dec. 14, 1992) (“Adams”)

I. INTRODUCTION

Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc. (collectively “Petitioner”) petition for *Inter Partes Review* (“IPR”) of claims 31 and 61 (“the Challenged Claims”) of U.S. Patent No. 6,233,389 (“the ’389 patent”). The ’389 patent is titled “Multimedia Time Warping System” and “relates to the real time capture, storage, and display of television broadcast signals.” SE1001, 1:6-9. The ’389 patent includes 61 claims. Most of those claims are directed to “television (TV) broadcast signals” and specific features related thereto (*see* SE1001, cl. 1-30, 32-60) and are not subject to this IPR petition.

This petition relates only to claims 31 and 61 of the ’389 patent. During prosecution of the application that resulted in the ’389 patent, claims 31 and 61 received a first action allowance. SE1002, 1007, 111. The stated reason for allowance merely recited all claim elements. SE1002, 111.

The ’389 patent also was reexamined twice. *See* SE1016; SE1017. The first reexamination challenged unrelated claims—not claims 31 and 61. SE1016, 1. In the second reexamination, claims 31 and 61 were challenged and ultimately confirmed patentable over the art cited in the reexamination request. SE1017, 1, 1336. The stated reason for allowance was that the cited references “do not teach or render obvious a transform object that automatically flow controls source and

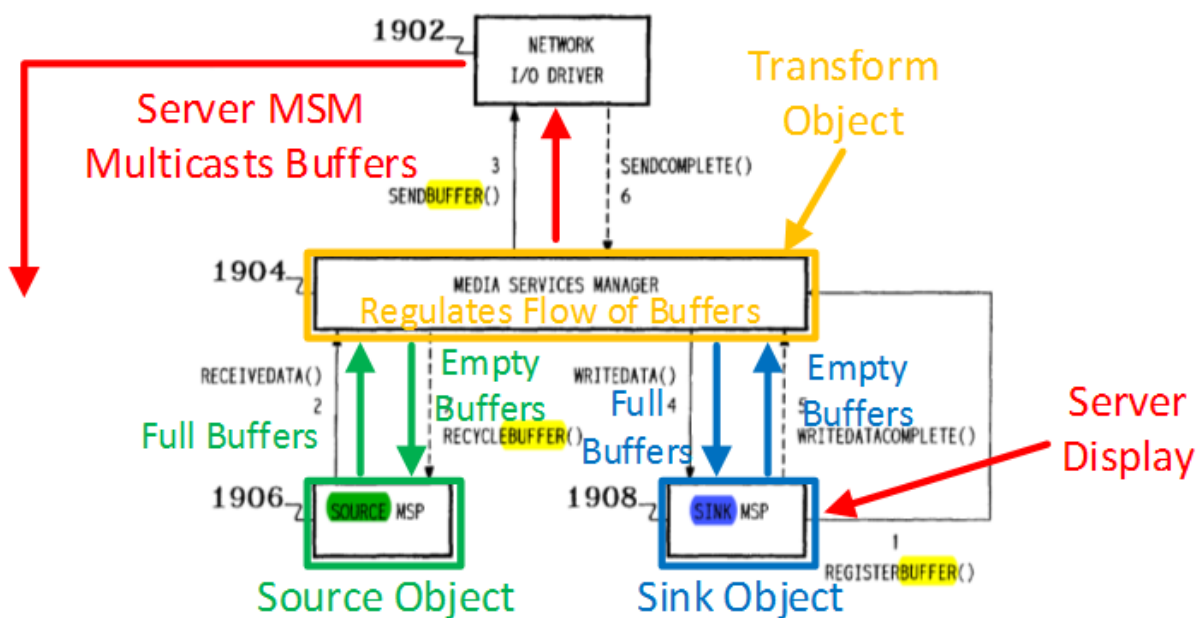
sink objects as claimed.” SE1017, 1340. In discussing automatic flow control by a transform object, the examiner credited expert testimony and explained:

This architecture is “self-regulating” with respect to the flow of data down the pipeline as noted at column 8, lines 47-51 and column 21, lines 24-25. In other words, the transform object is not merely self-regulating in an abstract sense. Rather, it ***controls the flow*** of data through the pipeline. (Villasenor at 7) Hence, the automatic flow control can be thought of as intelligent.

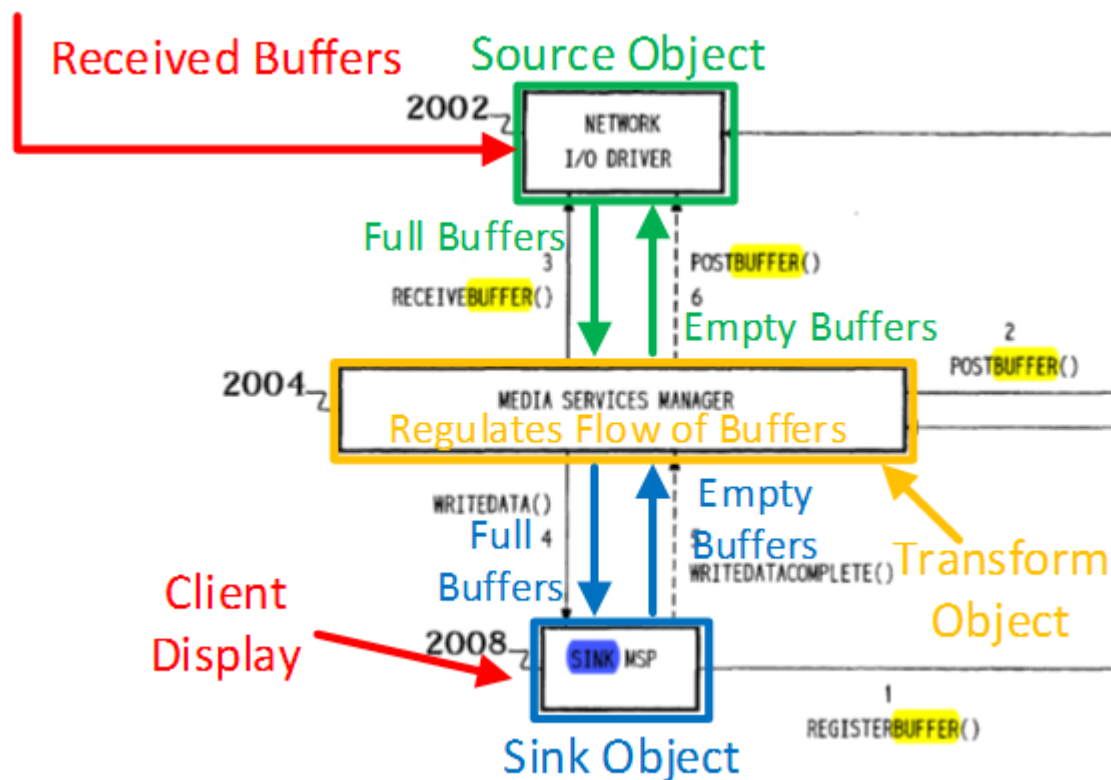
SE1017, 1336-1337 (emphasis added). The examiner then distinguished the cited references, stating: “Thomason just manages buffers in a reactive way in response to the flow. There is no disclosure in Thomason of the kind of intelligent management performed by the recited centralized transform object architecture as disclosed in the ’389 patent.” SE1017, 1339 (emphasis added).

As described throughout this petition, U.S. Patent 5,557,724 to Sampat (“Sampat”) describes a system with a centralized transform object, which intelligently controls the flow data. SE1004. For example, Sampat describes a “Media Services Manager” as its transform object, stating: “Media services manager (MSM) 1608 ***manages the flow of data*** through server software architecture 1512” by regulating “buffer” passing between the “source” and the “sink” objects. SE1004, 9:10-26, 9:57-10:7, 10:26-29, 14:43-54, 15:41-17:28, 18:28-54. Indeed, FIGS. 19 and 20 (reproduced below) illustrate “the ***flow*** of

data” through Sampat’s server and client, showing that each includes a central
“MSM” that automatically controls flow by passing “*source* buffers” and “*sink*
buffers.” SE1004, 17:8-67.



SE1001, FIG. 19 (annotated).



SE1001, FIG. 20 (annotated).

Therefore, Sampat discloses the only feature relied on by the CRU examiner in confirming claims 31 and 61. In fact, as detailed throughout this petition, Sampat discloses all elements of claims 31 and 61 of the '389 patent, including the claimed physical data source, and the source, sink, transform, and control objects. SE1004, SE1004, 2:10-16, 4:7-25, 7:59-8:22, 9:10-12, 9:37-53, 13:59-63, 14:33-54, 17:8-44, FIGS. 16-20; SE1003, ¶¶26-220.

Sampat was never cited during prosecution, was never cited during the first reexamination, and was not cited during the second reexamination *prior to* the Notice of Intent to Issue the Ex Parte Reexamination Certificate. See SE1002 and

SE1016, *passim*; *see* SE1017, 1-1352. Patent Owner attempted to cite Sampat as part of an IDS of one hundred new references *after* the Notice of Intent to Issue the Ex Parte Reexamination Certificate. SE1017, 1356. The IDS was deemed an “untimely paper” and was “expunged.” SE1017, 1357. Patent Owner attempted to file the IDS again, along with a petition. SE1017, 1359-1370, 1470-1474. The petition was dismissed and the IDS was not considered. SE1017, 1481. Patent Owner was invited to “file a new request for reexamination for consideration of such reference(s)” (SE1017, 1481), but Patent Owner never did so. Accordingly, Sampat was never considered during prosecution or reexamination, alone or in combination with the other references cited herein.

While Patent Owner never acted on the suggestion in the Decision On Petition to file a new request to have Sampat considered, Petitioner does so now in *inter partes* review and asks the Board to find the ’389 patent unpatentable as anticipated by Sampat, or as obvious over Sampat and the additional references cited below.

II. REQUIREMENTS FOR IPR

A. Grounds for Standing

Petitioner certifies that the ’389 patent is available for IPR. This petition is being filed within one year of service of a complaint against Petitioner on 9/8/2015. Petitioner is not barred or estopped from requesting review of the

Challenged Claims.

B. Challenge and Relief Requested

Petitioner requests IPR of the Challenged Claims on the grounds in the table below, as explained in this petition and in SE1003, the Declaration of John Michael Strawn, Ph.D.

Ground	'389 Claims	Basis
Ground 1	31 and 61	Anticipated by Sampat (client-side)
Ground 2	31 and 61	Obvious over Sampat (client -side), VfW, SoundBlaster, and Gerber
Ground 3	31 and 61	Anticipated by Sampat (server-side)
Ground 4	31 and 61	Obvious over Sampat (server-side), VfW, SoundBlaster, and Gerber

The earliest proclaimed priority date of the '389 patent is July 30, 1998 (the "Critical Date"). Each reference pre-dates this and qualifies as prior art:

Reference	Date	Section
Sampat	9/17/1996 (issued)	102(b)
SoundBlaster	1991 (published)	102(b)
VfW	1993 (published)	102(b)

Reference	Date	Section
Gerber	10/31/1996 (filed)	102(e)

SoundBlaster contains a copyright notice indicating publication in 1991. SE1006, 2; *Ford Motor Co. v. Cruise Control Techs. LLC*, IPR2014-00291, Paper 44 at p. 8 (PTAB June 29, 2015) (citing Joseph S. Dubin, *The Universal Copyright Convention*, 42 Cal. L. Rev. 89, 103 (1954)) (“a copyright notice bearing the symbol ‘©’ provides some evidence of a date of publication . . . [because] such markings generally indicate the ‘first year of publication.’”). This publication date is confirmed by an Amazon.com web listing also indicating publication in 1991—more than a year before the Critical Date. SE1019, 1; *see also* SE1004, 4:56-57 (identifying existence of SoundBlaster before 1993).

VfW contains a copyright notice indicating publication in 1992 and 1993. SE1007, 2; *Ford Motor Co.*, IPR2014-00291, Paper 44, p. 8 (PTAB June 29, 2015). This publication date is confirmed by Musser, which explains that Microsoft Video for Windows and the associated .zip file containing the Programmer’s Guide for Microsoft Video for Windows Development Kit (“VfW”) was publicly distributed by 1993. SE1020, 2; *see also* SE1021, 1-2; SE1004, 14:10-12 (identifying the “Microsoft Video for Windows” as existing by 1993).

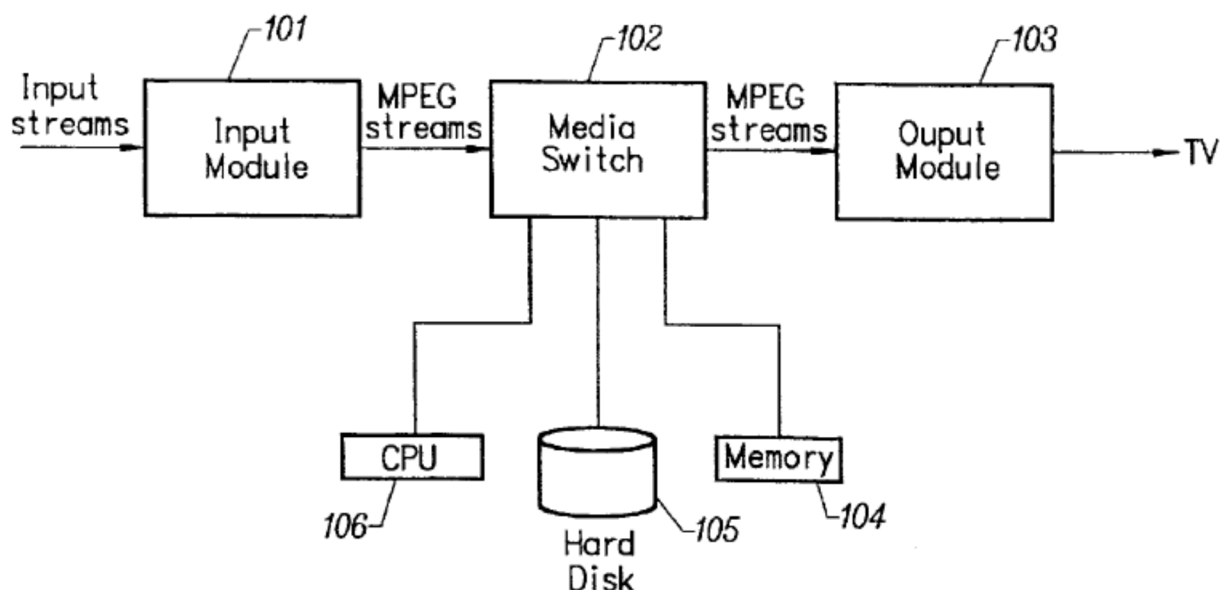
The publication date of VfW is also confirmed by Rhodes, a patent filed in 1993..

SE1008, 29:5-13.

III. BACKGROUND

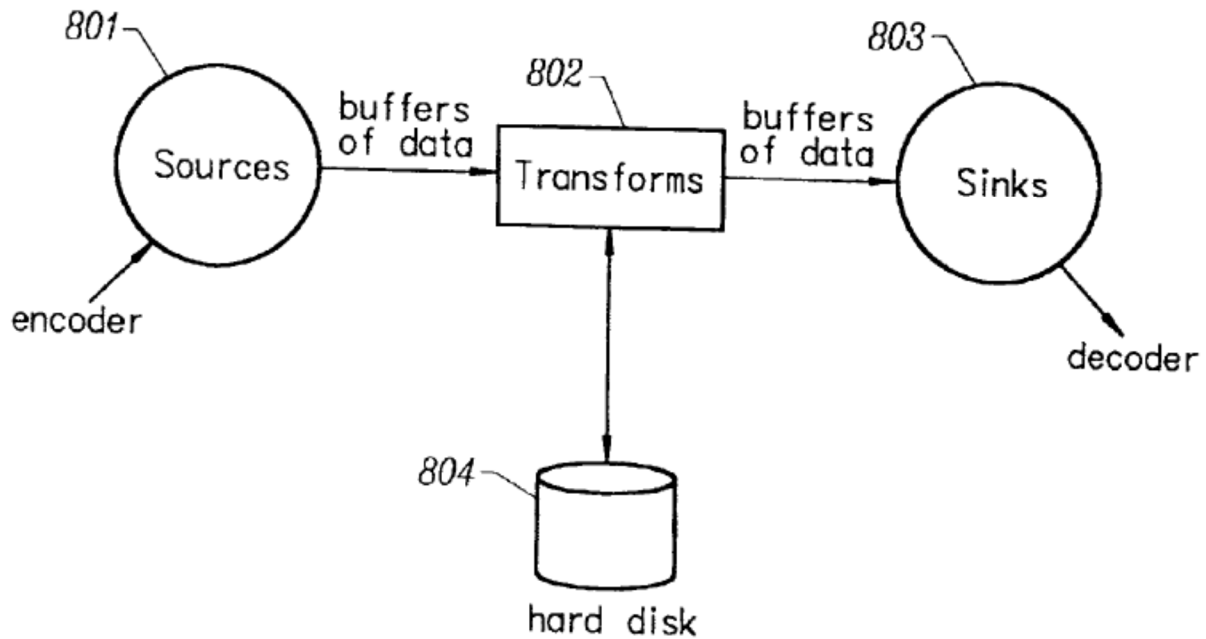
A. Overview of the '389 Patent

The '389 patent “relates to the real time capture, storage, and display of television broadcast signals.” SE1001, 1:6-9. FIG. 1 provides a “high level view” of the '389 patent’s system. SE1001, 2:44-45.



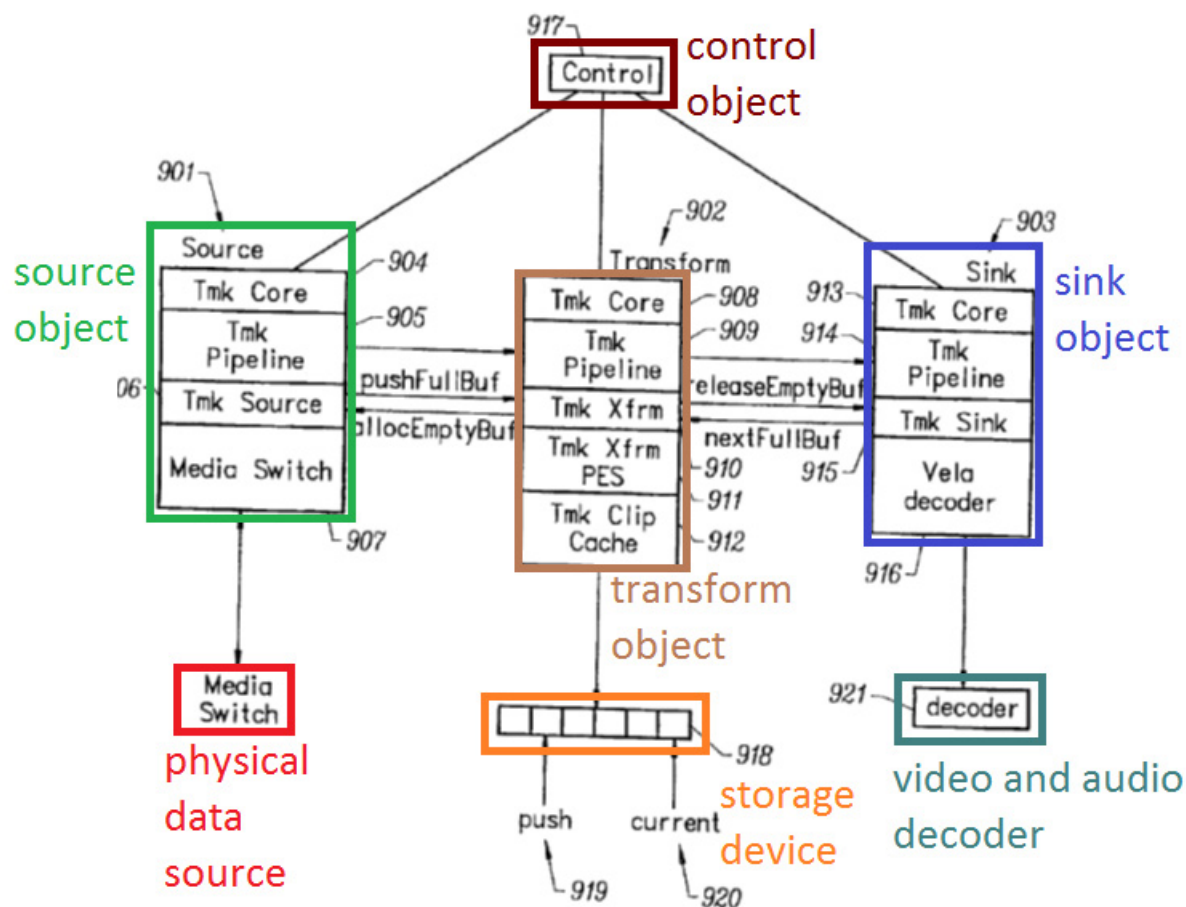
SE1001, FIG. 1.

Within that framework, claims 31 and 61 are directed to operations that control movement of data through the '389 patent’s system as performed by three conceptual components - Sources, Transforms, and Sinks.



SE1001, FIG. 8.

As shown below in FIG. 9, the '389 patent describes the use of a “source object” 901, “transform object” 902, and “sink object” 903. SE1001, 8:9-18, FIG. 9. A “control object” 917 accepts user commands. SE1001, 9:25-32.



SE1001, FIG. 9 (annotated).

The '389 patent describes the source object calling the transform object for a buffer to fill. SE1001, 8:45-48. The transform object provides the empty buffer to the source object and then takes the full buffer from the source object and stores it on a storage device. SE1001, 9:2-9. The sink object calls the transform object for a full buffer and then sends the data to a decoder. SE1001, 9:10-16. It then releases the empty buffer to the transform object for use again by the source object. SE1001, 8:55-59.

Under this system, the source object waits for the transform object to provide an empty buffer. Similarly, the sink object also waits for the transform object to provide a full buffer. According to the '389 patent, "[t]his means that the pipeline is self-regulating; it has automatic flow control." SE1001, 8:48-49.

B. Prior Art

Sampat discloses a system for processing data streams, which is represented at its highest level in Figure 1, showing input devices 108, 110, and 112, a server 102, and clients 104.

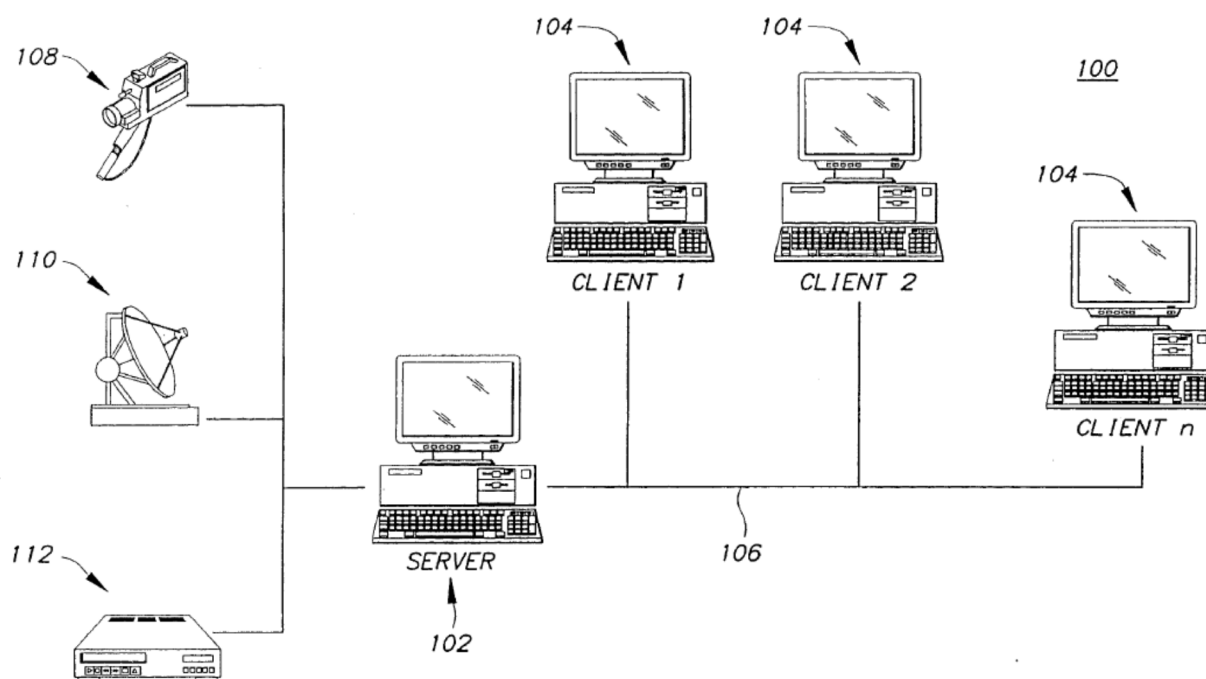
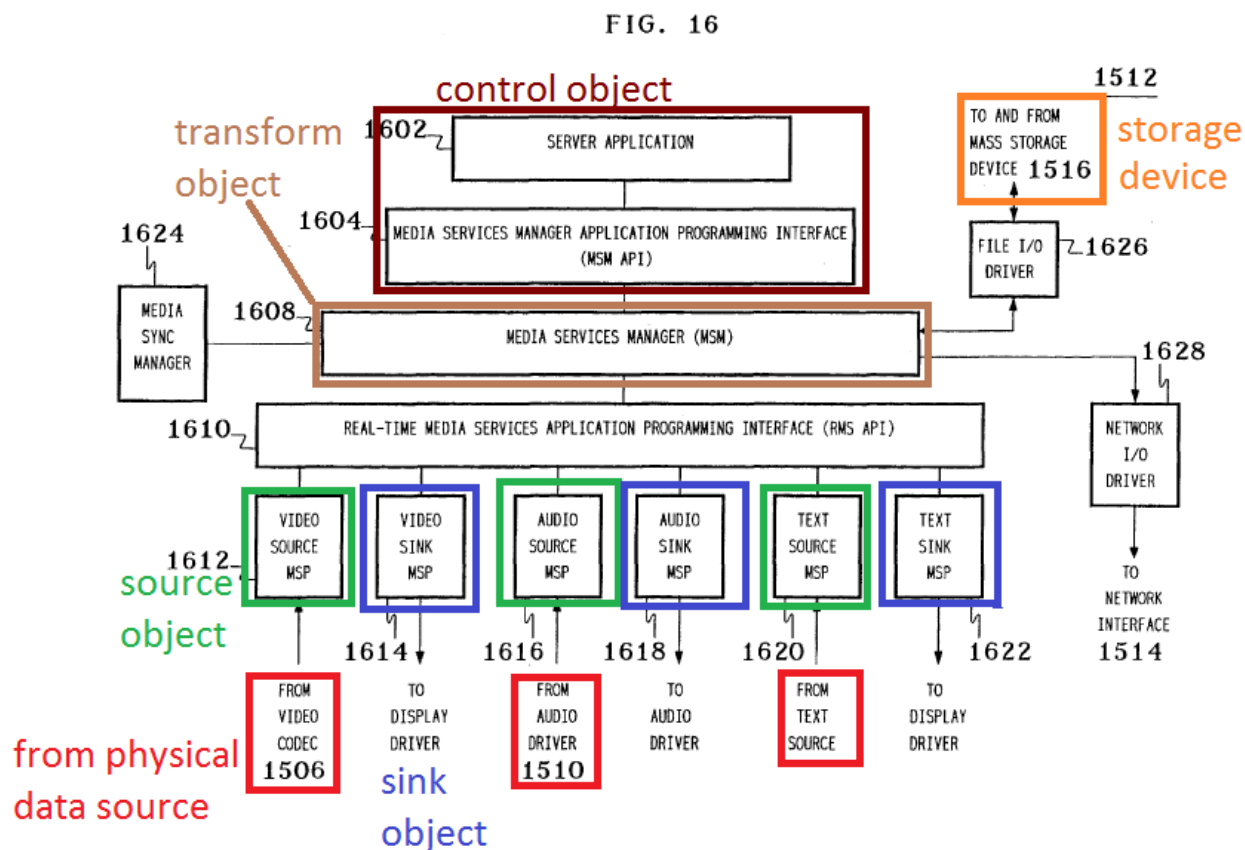


FIG. 1

SE1004, FIG. 1. Sampat discloses source, sink, and transform objects in both its server and its clients. For example, Figure 16 of Sampat (below) shows a source

object colored green (source MSPs), a sink object colored blue (sink MSPs), and a transform object colored brown (media services manager).



SE1004, FIG. 16 (annotated).

Sampat's transform object (MSM 1608) clearly performs automatic flow control, as Sampat explains: "Media services manager (MSM) 1608 *manages the flow of data* through server software architecture 1512." SE1004, 9:10-26 (emphasis added). In addition to automatic flow control, Sampat also discloses the features of server 102 and client 104 performing the other functional elements found in claims 31 and 61, as further described below.

To the extent that any element of claims 31 and 61 are found to not be *explicitly* described by Sampat, such features and functionality were conventional and well-known. For example, Sampat discloses that its preferred embodiment includes certain well-known components, including: “Intel® SmartVideo® Recorder (ISVR),” “SoundBlaster Pro from Creative Labs,” and “Microsoft Video for Windows.” SE1004, SE1004, 4:56-57, 8:25-27, 14:10-12. To the extent that Sampat does not describe one or more details of these devices, such details were well-known and described in contemporaneous publications. For example, Gerber describes the Intel Smart Video Recorder in further detail. SE1005, 5:10-16, 5:20-60, FIG. 3; SE1003, ¶¶31, 132. SoundBlaster describes details of the same SoundBlaster device preferred by Sampat. SE1006, 10, 20, 146, 152-153, 158; SE1003, ¶¶31, 79. VfW describes details of Microsoft Video for Windows. SE1007, 8, 157-211; SE1003, ¶¶31, 78.

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

The broadest reasonable construction/interpretation (“BRI”) is applied herein.¹ In prior court proceedings (where Petitioner was not a party), Patent

¹ The claim construction standard for district court (“ordinary and customary meaning”) is different than the BRI standard applied in IPR. Due to the different standards, disclosure of the references identified by Petitioner as teaching a claim

Owner proposed and a district court adopted constructions on over twenty terms from claims 31 and 61 of the '389 patent. *See* SE1011-SE1014 (construing some terms and applying plain meaning for others); *see also TiVo, Inc. v. EchoStar Commc'ns Corp.*, 516 F.3d 1290, 1307 (Fed. Cir. 2008) (endorsing the constructions of the appealed terms of the '389 patent as being “soundly based”). Accordingly, the BRI for these claim terms is *at least* as broad as the constructions proposed by Patent Owner and adopted—Patent Owner cannot credibly dispute this point. These terms include:

Claim Term	BRI Construction
Preamble	Not limiting. SE1014, 82; SE1001, 14:53-54, 17:3-4; SE1003, ¶36.
“Object”	“a collection of data and operations.” SE1014, 91-98; SE1001, 8:39-65, 9:23-47; SE1003, ¶36.

term of the '389 patent is not an admission that the claim term is met by any disclosure for infringement purposes, or that the claim term is enabled or meets the requirements for written description.

“source object”	“a collection of data and operations that (1) extracts video and audio data from a physical data source, (2) obtains a buffer [memory where data can be temporarily stored for transfer] from a transform object, (3) converts video data into data streams, and (4) fills the buffer [memory where data can be temporarily stored for transfer] with the streams.” SE1014, 91-98; SE1001, 8:39-51; SE1003, ¶36.
“sink object”	“a collection of data and operations that (1) obtains data stream buffers [memory where data can be temporarily stored for transfer] from a transform object and (2) outputs the streams to a video and audio decoder.” SE1014, 91-98; SE1001, 8:52-65; SE1003, ¶36.
“control object”	“a collection of data and operations that receives commands from a user that control the flow of broadcast data.” SE1014, 91-98; SE1001, 9:23-47; SE1003, ¶36.
“buffer”	“memory where data can be temporarily stored for transfer” SE1014, 9; SE1001, 2:16-17, 4:55-6:15; SE1003, ¶36.
“transform object”	“a collection of data and operations that transforms the form of data upon which it operates.”

	SE1014, 101-106; SE1001, 7:48-8:65; SE1003, ¶36.
“automatically flow controlled”	“self-regulated” SE1014, 101-106; SE1001, 8:19-65; SE1003, ¶36.
“wherein said source object is automatically flow controlled by said transform object”	“wherein said source object is self-regulated by said transform object” SE1014, 101-106; SE1001, 8:39-51; SE1003, ¶36.
“wherein said sink object is automatically flow controlled by said transform object”	“wherein said sink object is self-regulated by said transform object” SE1014, 101-106; SE1001, 8:52-65; SE1003, ¶36.
“wherein said source object extracts video	“wherein the source object obtains video and audio data from said physical data source” SE1014, 98-101; SE1001, 8:43-45; SE1003, ¶36.

and audio data from said physical data source”	
“control the flow of the broadcast data through the system”	“control the flow of the broadcast data within the system” SE1014, 109-111; SE1001, 8:19-65; SE1003, ¶36.
“wherein said control object sends flow command events to said source, transform, and sink objects”	“the control object sends information relating to a change of condition in the flow of the broadcast data to the source, transform and sink objects.” SE1013, 16-17; SE1001, 9:22-47; SE1003, ¶36.
“parses”	“analyzes” SE1014, 82-88; SE1001, 5:3-9, 6:37-46; SE1003, ¶36.
“parses video and audio data from	“analyzes video and audio data from the broadcast data” SE1014, 82-88; SE1001, 5:3-9, 6:37-46; SE1003, ¶36.

said broadcast data”	
“physical data source”	“hardware and software that parses video and audio data from said broadcast data” SE1014, 109-111; SE1001, 8:43-45; SE1003, ¶36.
“accepts broadcast data”	“accepts data that was transmitted” SE1014, 109-111, 3:30-61; SE1003, ¶36.

Claim terms not specifically construed above were interpreted under the BRI. SE1003, ¶38.

IV. APPLICATION OF PRIOR ART TO CHALLENGED CLAIMS

As detailed above (incorporated herein) and below, this request shows a reasonable likelihood that Petitioner will prevail on the Challenged Claims.

As mentioned above at Section II.A., claims 31 and 61 are primarily directed to software objects (i.e., source, sink, transform, and control objects) that control movement of data. Sampat discloses these claimed software objects in two components of its system: (1) server 102 and (2) client 104. Although operation of Sampat’s server 102 and operation of Sampat’s client 104 each meet the claimed features in a similar manner, they do so with different structure. As a consequence and for sake of clarity, Petitioner has addressed operation of Sampat’s server 102

in separate grounds from operation of Sampat's client 104. Specifically, this petition first addresses the patentability of claims 31 and 61 from the point of view of Sampat's client 104 (Grounds 1 and 2). SE1003 at ¶¶39-119. Then, this petition addresses the patentability of claims 31 and 61 from the point of view of Sampat's server 102 (Grounds 3 and 4). SE1003 at ¶¶120-217.

A. GROUND 1: Claims 31 and 61 are Anticipated under § 102(b) by Sampat (client-side)

Claims 31 and 61, and all of their claim elements, are taught by Sampat. As explained above, Ground 1 addresses Sampat when viewed from Sampat's client 104. For Ground 1, Figure 1 has been annotated below to schematically show that the source, sink, transform, and control objects are in the client 104, while the input devices and physical data source are upstream of the client 104.

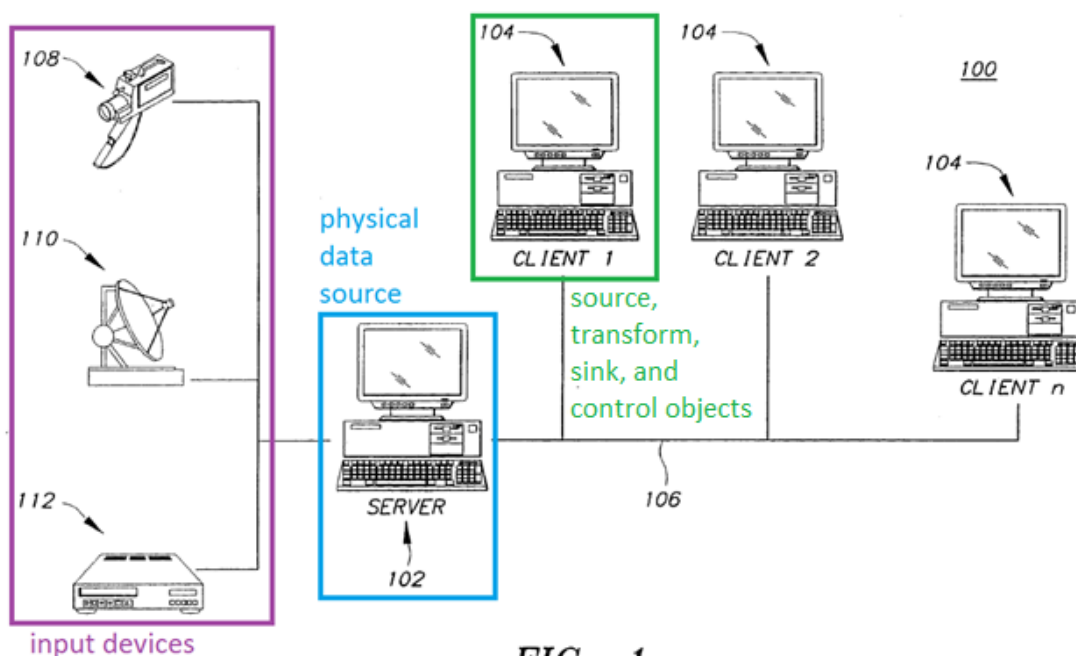
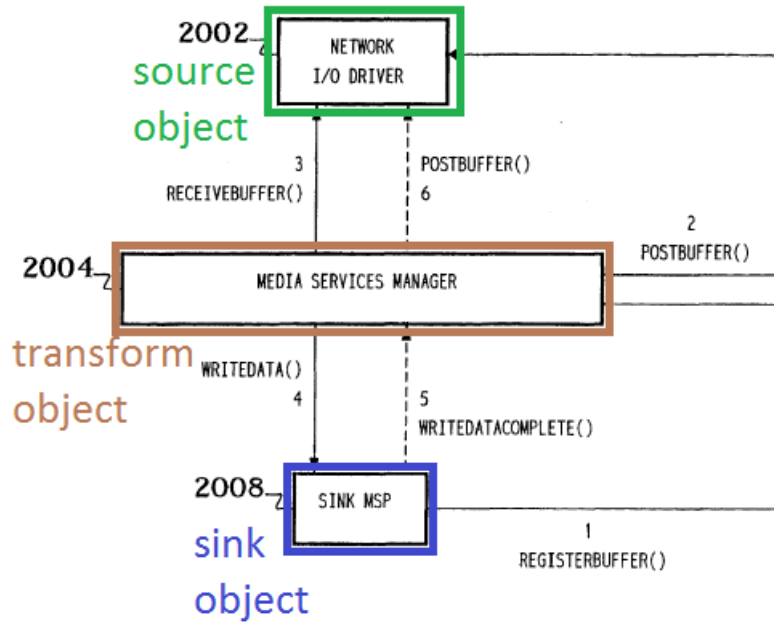


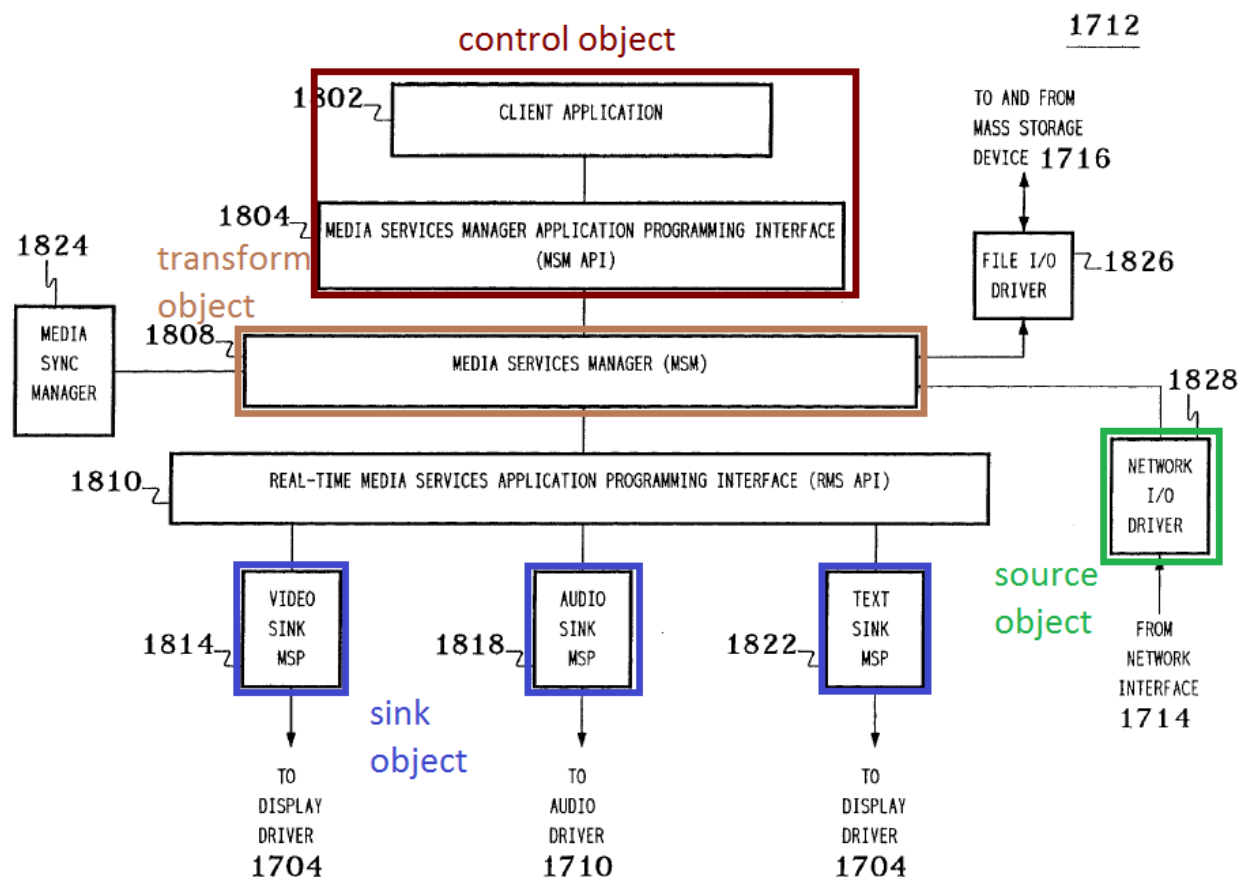
FIG. 1

SE1004, FIG. 1 (annotated for Ground 1). Figures 18 and 20, reproduced below, show the source, transform, sink, and control objects within the software architecture of client 104.



SE1004, FIG. 20 (annotated).

FIG. 18



SE1004, FIG. 18 (annotated). Claims 31 and 61 are identical, except that claim 31 is a process claim, which adds the word “providing” for five claim elements and also modifies the preamble slightly. Accordingly, claims 31 and 61 will be considered at the same time, with corresponding elements addressed together.

[31.Pre] A process for the simultaneous storage and play back of multimedia data, comprising the steps of:

[61.Pre] An apparatus for the simultaneous storage and play back of multimedia data, comprising:

The preamble of claims 31 and 61 are not limiting. A preamble is limiting only “when read in the context of the entire claim, recites limitations of the claim, or, if the claim preamble is ‘necessary to give life, meaning, and vitality’ to the claim.” *Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1305 (Fed. Cir. 1999). In this case, the preamble of claims 31 and 61 do not recite limitations of the claims and are not necessary to give life, meaning, or vitality to the claims—they can be understood without the preamble. Indeed, none of the claim elements have antecedents in the preamble. Moreover, Patent Owner has already “agreed that the preambles of Claims 31 and 61 of the ’389 patent are not limiting.” SE1014, 82. Accordingly, the preambles of claims 31 and 61 are not limiting.

Even if the preamble were limiting, Sampat discloses an apparatus and corresponding process for the simultaneous storage and play back of multimedia data. SE1004, 13:59-63 (“Client 104 also supports the recording of data from the network to mass storage device 1716 with or without concurrent playing of the multicast data”); *see also* 2:10-16. SE1003, ¶45.

While the preamble of claim 61 refers to “[a]n apparatus,” the term “apparatus” does not exclude an electrically-connected system. For example, claim element [61.n] refers to “the system”—showing that claim 61 uses the terms “apparatus” and “system” interchangeably. *See also* SE1014, 109-111 (construing that claim element to cover “control the flow of the broadcast data within *the*

system”) (emphasis added). Similarly, claim 54 (which is not in dispute here) claims an “apparatus” which includes the separate component of a VCR. SE1001, 17:7-12. Accordingly, even if the preamble of claim 61 were limiting, which it is not, Sampat’s multicast system 100 is an “apparatus” and a “system” within the meaning of claim 61, especially under BRI.

[31.a] providing² [61.a] a physical data source,

Sampat discloses providing a physical data source. SE1003, ¶¶46-56. The BRI of “physical data source” includes “hardware and software that parses video and audio data from said broadcast data.” SE1014, 109-111; SE1001, 8:43-45. The BRI of “parses video and audio data from said broadcast data” includes “analyzes video and audio data from the broadcast data.” SE1014, 82-88; SE1001, 5:3-9, 6:37-46.

As viewed from the client-side, Sampat discloses a physical data source in the form of server 102, which includes hardware and software. SE1004, 4:7-25. Figure 1, reproduced below, shows the physical data source (blue).

² The term “providing” appears only in claim 31, with all other terms common.

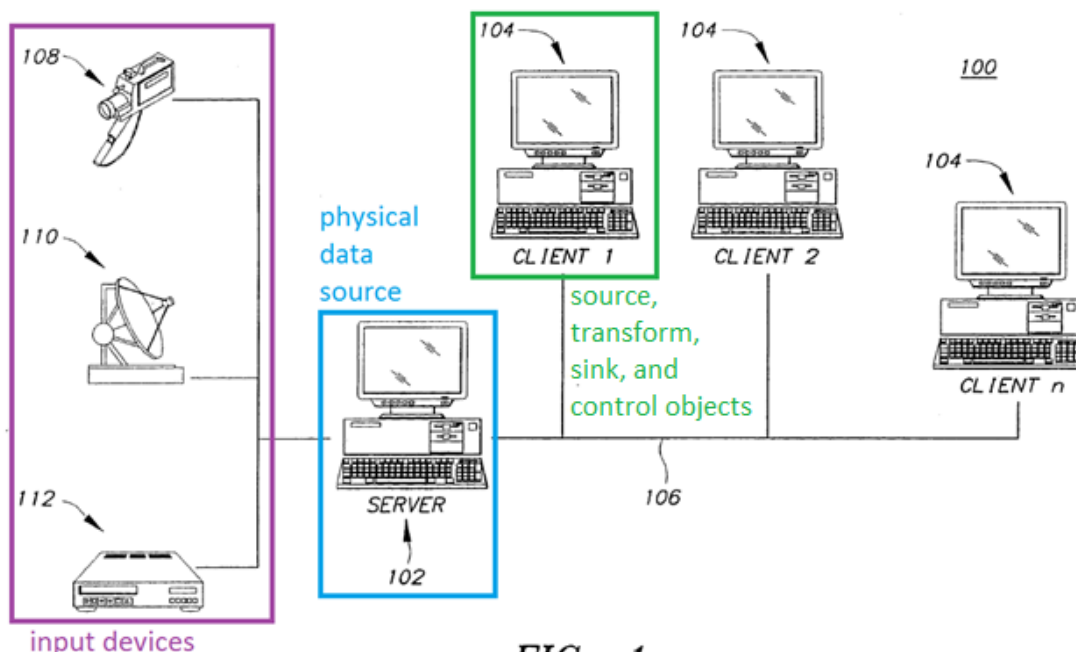


FIG. 1

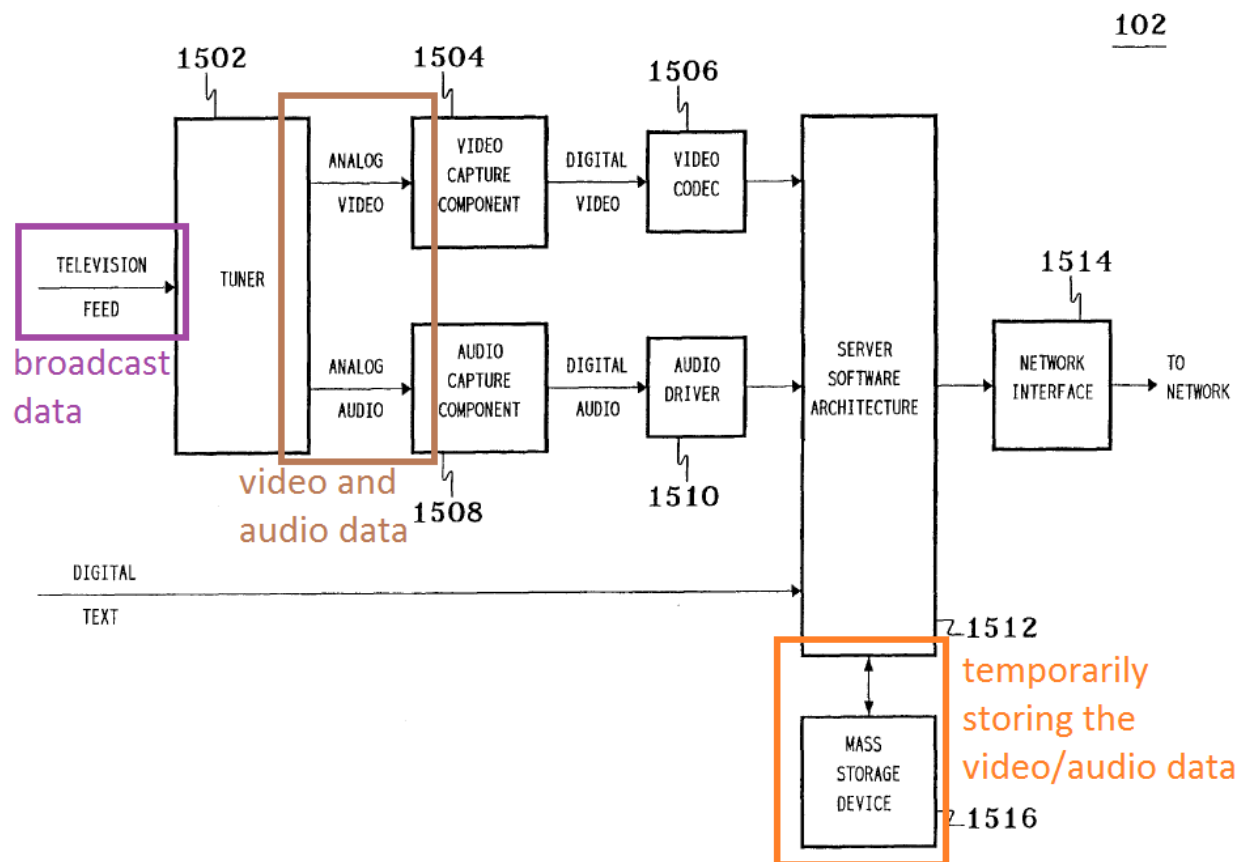
SE1004, FIG. 1 (annotated); SE1003, ¶48.

Sampat discloses server 102 (the physical data source) accepting broadcast data from one or more input devices (e.g., camera 108, antenna 110, and VCR 112). SE1004, 4:7-14 (“Server 102 is capable of capturing analog audio and video signals from three different sources: (1) signals generated locally by camera 108, (2) signals received by antenna 110 from a remote source, and (3) recorded signals from VCR 112”), 38:42-54 (the system can be used with “broadcasting”). “For example, server 102 may receive via antenna 110 a first television program” and “may receive a second television program ... from VCR 112.” *Id.*, 4:15-23; *see also* SE1001, 3:32-43 (describing receiving broadcast data from a “TV source

transmitter”), 11:50-51, FIG. 13 (describing and showing a VCR as another input device); SE1003, ¶49.

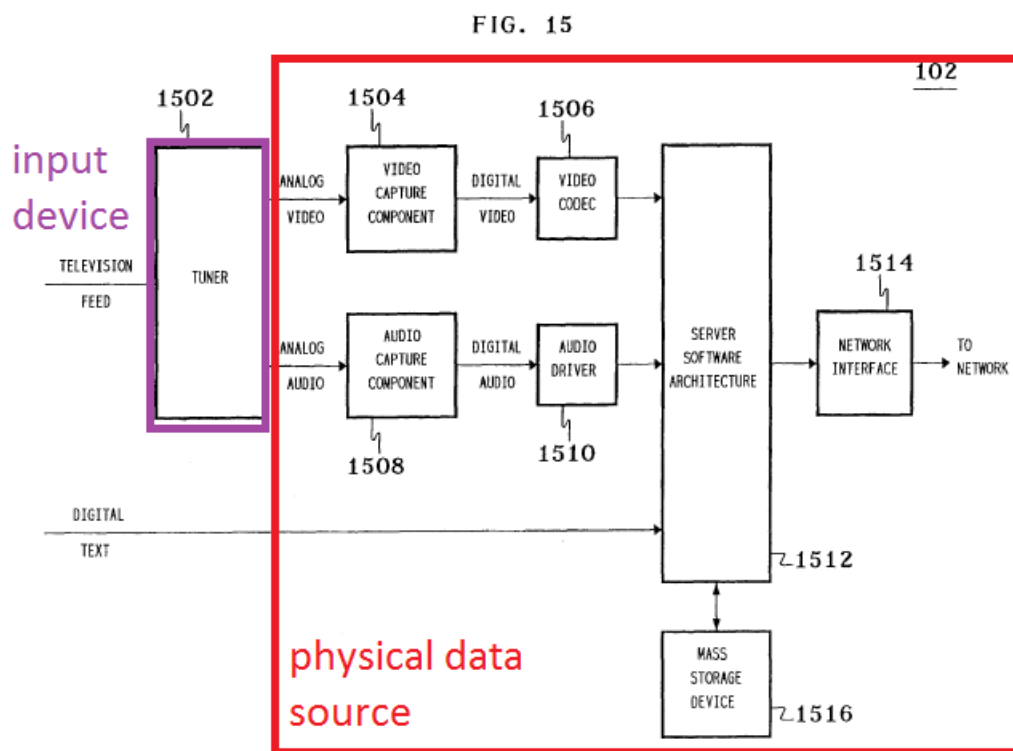
Sampat further discloses server 102 parsing video and audio data from the broadcast data. SE1004, 7:55-63 (“In particular, tuner 1502 of server subsystem 102 receives, demodulates, and splits one or more analog television feed signals into their constituent analog audio and video signals.”); 4:24-43 (describing the server 102 digitizing and fragmenting the received signals). Figure 15, reproduced below, shows a block diagram of server 102, with its tuner 1502 parsing the broadcast data (purple) into video and audio data (brown).

FIG. 15



SE1004, FIG. 15 (annotated); SE1003, ¶50.

Alternatively, Sampat discloses a physical data source (hardware and software) as the server 102 *without* the tuner 1502. In that case, the tuner 1502 is the input device, as shown below.



SE1004, FIG. 15 (annotated-alternative); SE1003, ¶¶53-54.

Such alternative interpretations are consistent with Patent Owner's infringement contentions. *See* SE1015, 39-48, 101-110. For example, Patent Owner alleges that the "Samsung DVR has an input device that may include cable input, tuners..." (SE1015, 39), which is consistent with Sampat's disclosure of tuner 1502 (SE1004, 7:59-61, 8:22-25): SE1003, ¶55.

Sampat also discloses that this physical data source *parses* video and audio data under the BRI, which includes "*analyzes* video and audio data from said broadcast data." SE1014, 82-88. The server 102 analyzes video and audio data from the broadcast data in multiple ways. For example, "[v]ideo capture

component 1504 captures and converts the analog video signals into digital video data streams” and “audio capture component 1508 captures and converts the analog audio signals into digital audio data streams,” which includes analysis of video and audio data when converting into digital video. SE1004, 7:62-65; SE1003, ¶56. Additionally, “[v]ideo codec 1506 compresses the digital video data streams” and “[a]udio driver 1510 places the audio data into buffers,” which also includes analysis of video and audio data as broadly construed. SE1004, 8:7-10; SE1003, ¶56. Moreover, the server software architecture 1512 further analyzes the video and audio data. SE1004, 8:12-17, *see also* 8:50-13:33. Accordingly, Sampat also discloses the claimed physical data source under this alternative interpretation. SE1003, ¶¶53-56.

[31.b, 61.b] wherein said physical data source accepts broadcast data from an input device, parses video and audio data from said broadcast data, and temporarily stores said video and audio data;

Sampat discloses that the physical data source accepts broadcast data from an input device, parses video and audio data from said broadcast data, and temporarily stores said video and audio data. As explained in claim element [31.a], Sampat discloses that the physical data source accepts broadcast data from an input device and parses video and audio data from said broadcast data. SE1003, ¶¶46-56; SE1004, 4:7-43, 7:55-63, 38:42-54, FIGS. 1, 15.

Sampat further discloses the server 102 temporarily storing the video and audio data. SE1004, 9:10-26 (describing temporary storage for later multicasting or processing); SE1003, ¶52. This temporary storage is also illustrated in Figure 15, shown above. Server software architecture 1512 of server 102 is shown in Figure 15 temporarily storing and retrieving the video and audio data to mass storage device 1516 (shown in orange). SE1004, FIG. 15, 9:10-26; SE1003, ¶¶50, 52.

[31.c] providing [61.c] a source object,

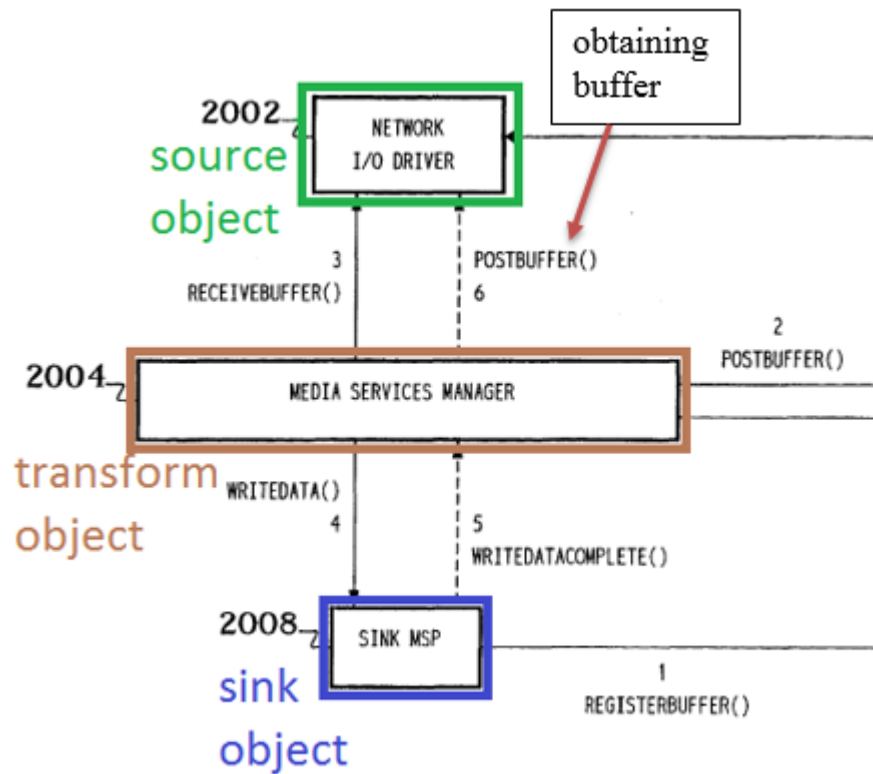
Sampat discloses providing a source object. SE1003, ¶¶57-64. The BRI for “source object” includes “a collection of data and operations that (1) extracts video and audio data from a physical data source, (2) obtains a buffer [memory where data can be temporarily stored for transfer] from a transform object, (3) converts video data into data streams, and (4) fills the buffer [memory where data can be temporarily stored for transfer] with the streams.” SE1014, 91-98; SE1001, 8:39-51.

As viewed from the client-side, Sampat describes the client 104 having a source object in the form of the network I/O driver (referred to as network I/O driver 1828 in FIG. 18, network I/O driver 2002 in FIG. 20, and network I/O driver 2100 in FIG. 21). SE1004, 14:38-43, 17:39-44, 18:9-16. Network I/O driver is a collection of data and operations. SE1004, 18:7-19:24; 19:24-28:62. Network I/O

driver also performs each of the four functions defined as being performed by a source object. SE1003, ¶¶59-63.

First, the network I/O driver (the source object) extracts video and audio data from the server 102 (the physical data source), which it obtains via the network interface 1714. SE1004, 14:38-43. This meets the BRI of “wherein said source object extracts video and audio data from said physical data source,” which includes “wherein the source object obtains video and audio data from said physical data source” SE1014, 98-101; SE1003, ¶60.

Second, Sampat describes the source object (network I/O driver) obtaining a buffer from the transform object (MSM), stating: “MSM 2004 re-posts the buffer to network I/O driver 2002 to be reused.” SE1004, 17:64-67, *see also* 17:50-59. The source object (network I/O driver) obtaining a buffer from the transform object (MSM) is also shown in Figure 20:



SE1004, FIG. 20 (annotated), *see also* 17:50-59, 17:64-67; SE1003, ¶61.

Third, Sampat discloses that the source object (network I/O driver) converts video data packets into data streams: “In a client, network I/O driver 2100 receives related, time-stamped data *packets* from the network and transmits data *streams* corresponding to those data packets to the client media services manager for display and/or recording of the multicast channel data.” SE1004, 18:20-25 (emphasis added), *see also* 18:59-63; SE1003, ¶62.

Fourth, Sampat discloses that the source object (network I/O driver) fills the buffer with data streams, stating: “When data is received by network I/O driver 2002 from the network, network I/O driver 2002 *fills a sink buffer* and passes it to

MSM 2004.” SE1004, 17:55-58 (emphasis added). Dr. Strawn confirms that a POSITA would have understood the network I/O driver in Sampat’s client 104 to include a source object that performs all of these claimed functions. SE1003, ¶63.

[31.d, 61.d] wherein said source object extracts video and audio data from said physical data source;

As described above at [31.c, 61.c], Sampat discloses this wherein clause, which is identical to language in the construction of source object as addressed. SE1003, ¶60.

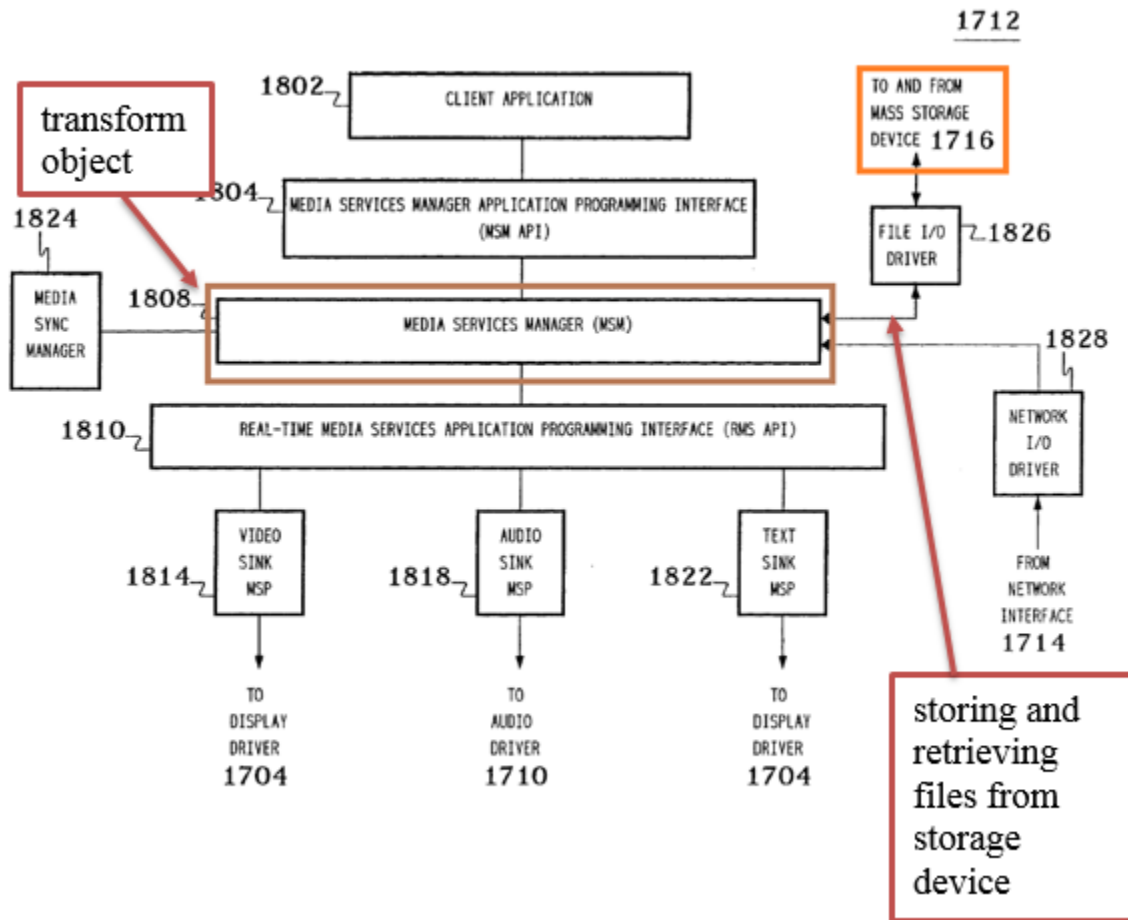
[31.e] providing [61.e] a transform object,

Sampat discloses providing a transform object. SE1003, ¶¶65-71. Under BRI, a “transform object” is “a collection of data and operations that transforms the form of data upon which it operates.” SE1014, 101-106; SE1001, 7:48-8:65. The ’389 patent describes one type of transform as a ***temporal*** transform. SE1001, 7:64-8:8 (describing a transform as “temporal” when it “writes the buffer to a file 804 on the storage medium” and then pulls the buffer out “at a later time”); SE1003, ¶66.

Sampat describes a transform object in the form of the “media services manager” or “MSM” (MSM 1808 in Figure 18; MSM 2004 in Figure 20); SE1003, ¶67. SE1004, 14:33-54, 17:39-67, FIGS. 18 and 20. The MSM is a collection of data and operations. SE1004, 14:30-16:58, 17:39-18:5, FIGS. 18, 20, 24.

Sampat describes the MSM performing a *temporal* transform. SE1003, ¶68. Sampat's MSM performs a temporal transform by writing data streams to the mass storage device 1716 and retrieving the data streams at a later time for playback. SE1004, 15:7-9 ("MSM 1808 uses file I/O driver 1826 to store and retrieve data to and from mass storage device 1716."), 14:45-54, FIGS. 17-18. Sampat describes the MSM temporarily storing and retrieving data flowing "through MSM 1808...[f]rom the network to mass storage device 1716 (for recording of multicast data for subsequent processing), and [f]rom mass storage device 1716 to a sink MSP (for playing of locally recorded multicast data)." SE1004, 14:45-54. In Figure 18, the MSM 1808 is shown transferring data streams "to and from mass storage device 1716" (quoted text colored orange) via the file I/O driver 1826. SE1004, FIG. 18; *see also* FIG. 17.

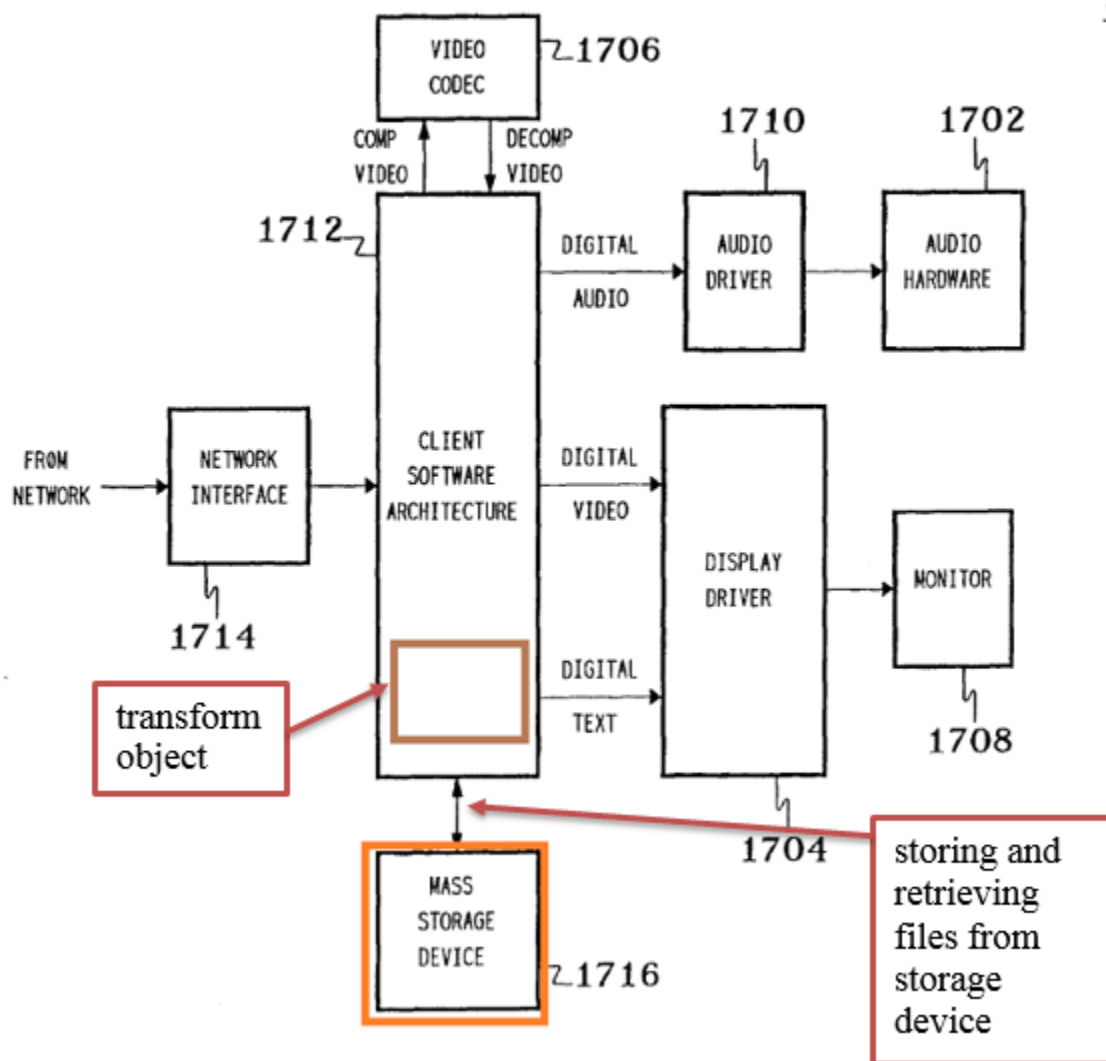
FIG. 18



SE1004, FIG. 18 (annotated); SE1003, ¶68.

FIG. 17

104



SE1004, FIG. 17 (annotated); SE1003, ¶¶68.

The temporal transform performed by Sampat's MSM is consistent with the '389 patent's temporal transform. See SE1001, 7:64-8:8. Thus, Sampat discloses a transform object that transforms the form of data upon which it operates in a similar manner to the '389 patent. SE1003, ¶¶69-70.

[31.f, 61.f] wherein said transform object stores and retrieves data streams onto a storage device;

As described above at [31.e, 61.e], Sampat's MSM performs a temporal transform by writing data streams to the mass storage device 1716 and retrieving the data streams at a later time for playback. SE1004, 14:45-54; 15:7-9; FIGS. 17-18. Indeed, the annotated version of Sampat's FIG. 18 in [31.e, 61.e] shows MSM 1808 transferring data streams "to and from mass storage device 1716." *Id.* By writing/retrieving data streams to/from a mass storage device, Sampat's transform object (MSM) stores and retrieves data streams onto a storage device. SE1003, ¶¶69-70.

[31.g, 61.g] wherein said source object obtains a buffer from said transform object, said source object converts video data into data streams and fills said buffer with said streams;

As described above at [31.c, 61.c], Sampat discloses this wherein clause, which is identical to language in the construction of source object as addressed. SE1003, ¶¶61-63.

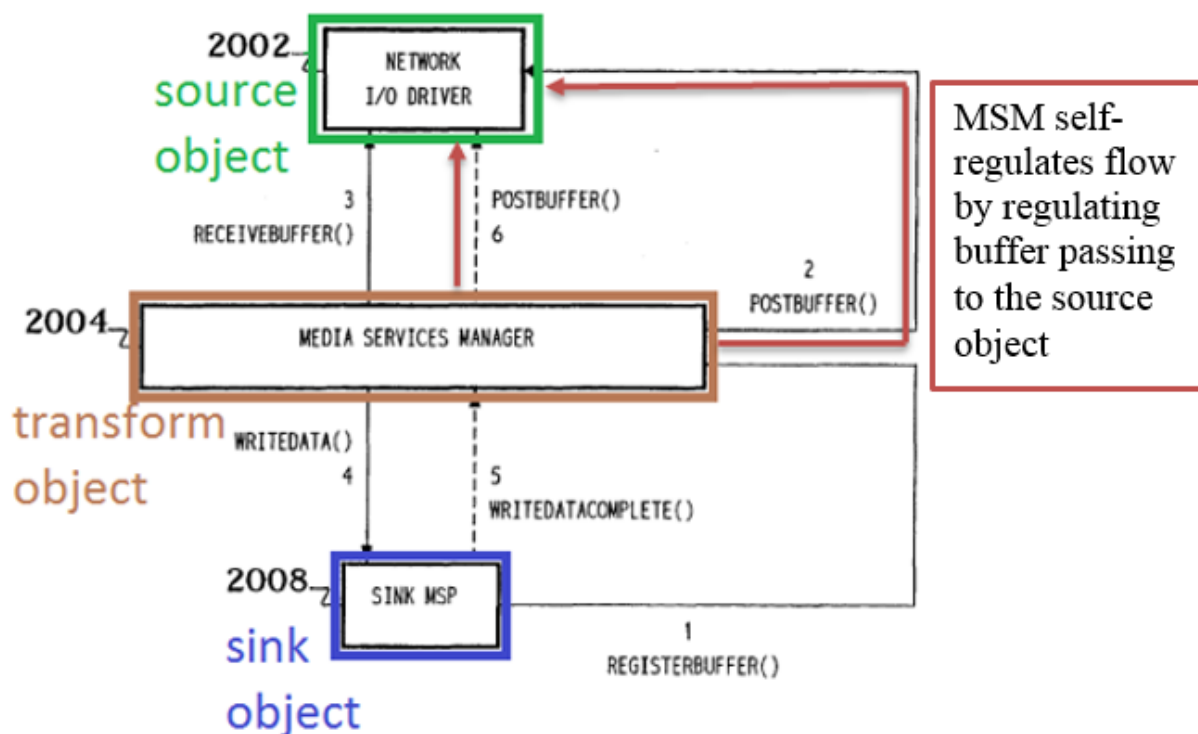
[31.h, 61.h] wherein said source object is automatically flow controlled by said transform object;

Sampat discloses that the source object is automatically flow controlled by said transform object. SE1003, ¶¶82-89. Under BRI, "automatically flow controlled" includes "self-regulated" and, consequently, this claim element includes "wherein said source object is *self-regulated* by said transform object."

SE1014, 101, 106 (emphasis added). District courts have repeatedly reached this construction and agreed that the analysis in the second reexamination of the '389 patent “did not limit or redefine” this relatively broad construction of “automatically flow controlled.” SE1014, 106; SE1013, 12. One district court explained: “TiVo merely reiterated during reexamination that *rather than the source object and sink object being merely reactive to the data flow, they are flow controlled by the transform object.*” SE1014 at 106 (emphasis added). Accordingly, the BRI of this claim term covers being “self-regulated” by the transform object. SE1003, ¶83.

In Sampat, the MSM (transform object) self-regulates flow, including flow from the network I/O driver (source object). SE1004, 14:43-54. Sampat explains: “*MSM 1808 manages the flow* of data through the client software architecture” including “[f]rom the network to a sink media service provider (MSP)” and “[f]rom the network to a mass storage device.” SE1004, 14:43-54 (emphasis added); SE1003, ¶84. The MSM regulates the flow of data from the source object both by starting and stopping flow (SE1004, 18:28-54, *see also* 17:50, 15:48-50) and also by regulating buffer passing during flow (SE1004, 17:39-18:5). Sampat shows and describes its transform object (MSM) controlling flow in the same way as described in the '389 patent, with the source object (network I/O driver) filling

buffers only when received from the MSM. SE1004 at 17:39-18:5); FIG. 20.



SE1004, FIG. 20 (annotated); SE1003, ¶87. For example, in step 2, Sampat explains: “When MSM 2004 initializes network I/O driver 2002, the MSM specifies the data streams to be received (i.e., which sink MSPs are open). MSM 2004 then posts all of the appropriate sink buffers to the network (using the MSM API function PostBuffer).” SE1004, 17:50-54; *see also* 17:64-67 (“MSM 2004 re-posts the buffer to network I/O driver 2002 to be reused (using the PostBuffer function)”). This description in Sampat is consistent with the description of automatic flow control in the ’389 patent, which states:

To obtain the buffer, the source object 901 asks the down stream object in his pipeline for a buffer (allocEmptyBuf). The source object 901 is blocked until there is sufficient memory. ***This means that the pipeline is self-regulating; it has automatic flow control.***

SE1001 at 8:45-51; *see also* 8:19-9:16. Accordingly, a POSITA would have understood that Sampat's MSM (transform object) automatically controls the flow of Sampat's network I/O driver (source object), not that the flow is merely reactive and independent of the MSM. SE1003, ¶¶88-89.

[31.i] providing [61.i] a sink object,

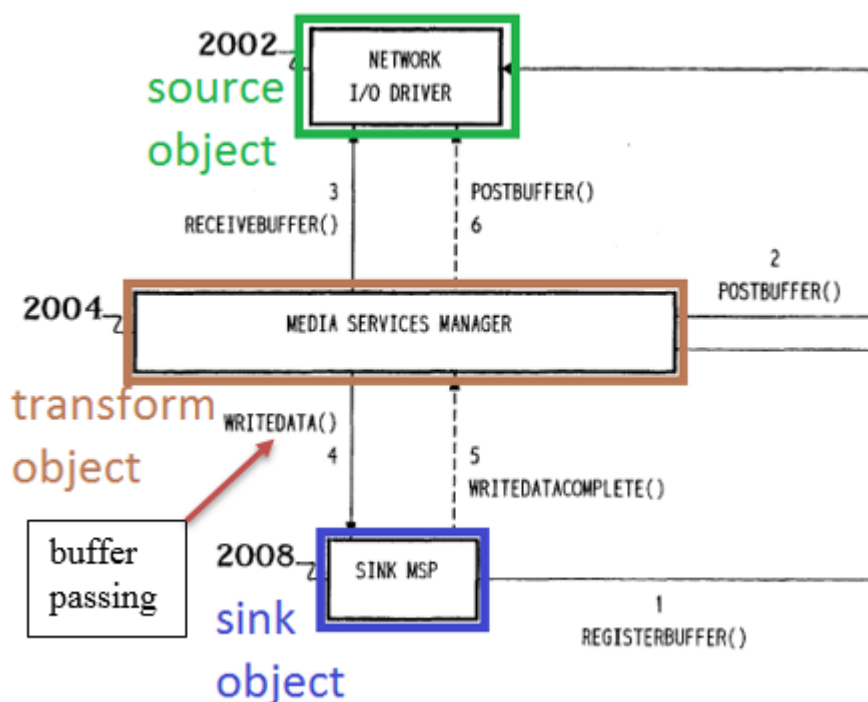
Sampat discloses providing a sink object. SE1003, ¶¶72-81. Under BRI, "sink object" includes "a collection of data and operations that (1) obtains data stream buffers [memory where data can be temporarily stored for transfer] from a transform object and (2) outputs the streams to a video and audio decoder."

SE1014, 91-98; SE1001, 8:52-65.

Sampat describes a sink object in the form of three sink MSPs: video sink MSP 1814, text sink MSP 1822, and audio sink MSP 1818. SE1004, 14:64-15:3. These sink MSPs are a collection of data and operations. SE1004 at 14:64-15:3, 17:39-67. These sink MSPs also performs each of the two functions defined as being performed by a sink object. SE1003, ¶74.

First, Sampat explains that the sink MSPs (sink object) obtain data stream buffers from the MSM (transform object), stating: "When data is received by

network I/O driver 2002 from the network, network I/O driver 2002 fills a sink buffer and passes it to MSM 2004” and that “MSM 2004 then writes the sink buffer data *to the sink MSP* that owns the buffer.” SE1004, 17:55-60. This buffer passing between the MSM and the sink MSP is shown in FIG. 20, reproduced below.



SE1004, FIG. 20 (annotated); *see also* 14:64-15:3 (describing all three sink MSPs receiving a “data stream from MSM 1808”). Dr. Strawn confirms that a POSITA would have understood that the sink MSPs receive the data stream buffers from the MSM. SE1003, ¶¶75-76.

Second, Sampat also teaches that the sink MSPs output the data streams to a video and audio decoder. Sampat explains:

Video sink MSP 1814 and text sink MSP 1822 receive a video data stream and a text data stream, respectively, from MSM 1808 and ***transmits the video and text data to display driver 1704*** of FIG. 17 for display on monitor 1708. Similarly, audio sink MSP 1818 receives an audio data stream from MSM 1808 and ***transmits the audio data to audio driver 1710 for play on audio hardware 1702***.

SE1004, 14:64-15:3 (emphasis added).

A POSITA would have understood that the display driver 1704 includes a video decoder. SE1003, ¶78. Such drivers were known to include decoders to convert signals for reproducing via a display. SE1003, ¶78. Sampat identifies “Microsoft Video for Windows” as an example of the display driver 1704.

SE1004, 14:6-12. Microsoft Video for Windows was well-known to a POSITA prior to the 1998 filing date of the ’389 patent, and was also known to include decoders. SE1003, ¶78; *see, e.g.*, SE1007, 8 (describing Video for Windows having “video compression and decompression drivers”), 157-211 (video decompression); SE1021, 2 (stating that “Video for Windows” display driver was well-known to include “three compressor-decompressor (codec) algorithms”); SE1008, 2:10-40 (describing “Video for Windows” receiving compressed video data and decoding/decompressing it).

A POSITA also would have understood that the system of Sampat includes an audio decoder. SE1003, ¶79. For example, Sampat explains that “clients 104

are preferably sound enabled with a SoundBlaster Pro from Creative Labs.”

SE1004, 4:56-57. Accordingly, Sampat’s disclosure of the sink object outputting to audio driver 1710 and audio hardware 1702 would output to the SoundBlaster driver and hardware. SE1004, 4:56-57, 15:1-15:3; SE1003, ¶79. Such SoundBlaster Pro devices were well-known to a POSITA prior to the 1998 filing date of the ’389 patent, and were also known to include audio decoders. SE1003, ¶79; *see, e.g.*, SE1006, 146, 152-153, 10.³

Under the BRI, Sampat’s decoding of audio and video via separate audio and video decoders is consistent with the ’389 patent. SE1001 at 4:23-29 (“separate decoders are used to convert MPEG elements back into audio or video analog components”). According to Patent Owner’s infringement contentions, Patent Owner contends that these claims cover a system with separate audio and video decoders. SE1015, 60 (alleging infringement by device with “two high definition [] video decoders, two advanced –audio decoders...”), *see also* 56-60. Accordingly, Sampat teaches the sink object outputting data streams to a video and

³ The exhibits SE1006, SE1007, and SE1021 are offered in Ground 1 only to confirm what was well-known about Microsoft Video for Window and SoundBlaster Pro—not that these references should be combined to show obviousness in Ground 1.

audio decoder. SE1003, ¶80.

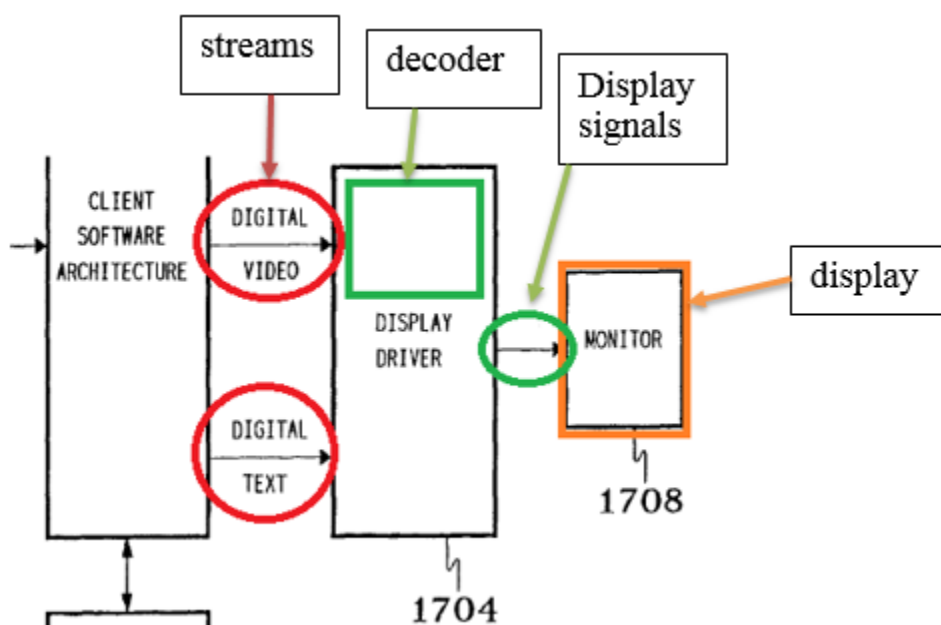
[31.j, 61.j] wherein said sink object obtains data stream buffers from said transform object and outputs said streams to a video and audio decoder;

As described above at [31.i, 61.i], Sampat discloses this wherein clause, which is identical to language in the construction of sink object as addressed.

SE1003, ¶¶75-80.

[31.k, 61.k] wherein said decoder converts said streams into display signals and sends said signals to a display;

Sampat discloses that the decoder converts the streams into display signals and sends the signals to a display. SE1003, ¶¶95-97. Sampat explains “Video sink MSP 1814 and text sink MSP 1822 receive a video data stream and a text data stream, respectively, from MSM 1808 and ***transmits the video and text data to display driver 1704 of FIG. 17 for display on monitor 1708.***” SE1004, 14:64-67 (emphasis added). As explained above at [31.i], the display driver 1704 is preferably Video for Windows, which includes a decoder. SE1004, 14:6-12; SE1007, 8, 157-211; SE1003, ¶¶78, 96. Figure 17 of Sampat also shows the display driver 1704 (which includes the decoder) converting streams into display signals and sending the signals to the display (monitor 1708). SE1003, ¶96.



SE1004, FIG. 17 (annotated).

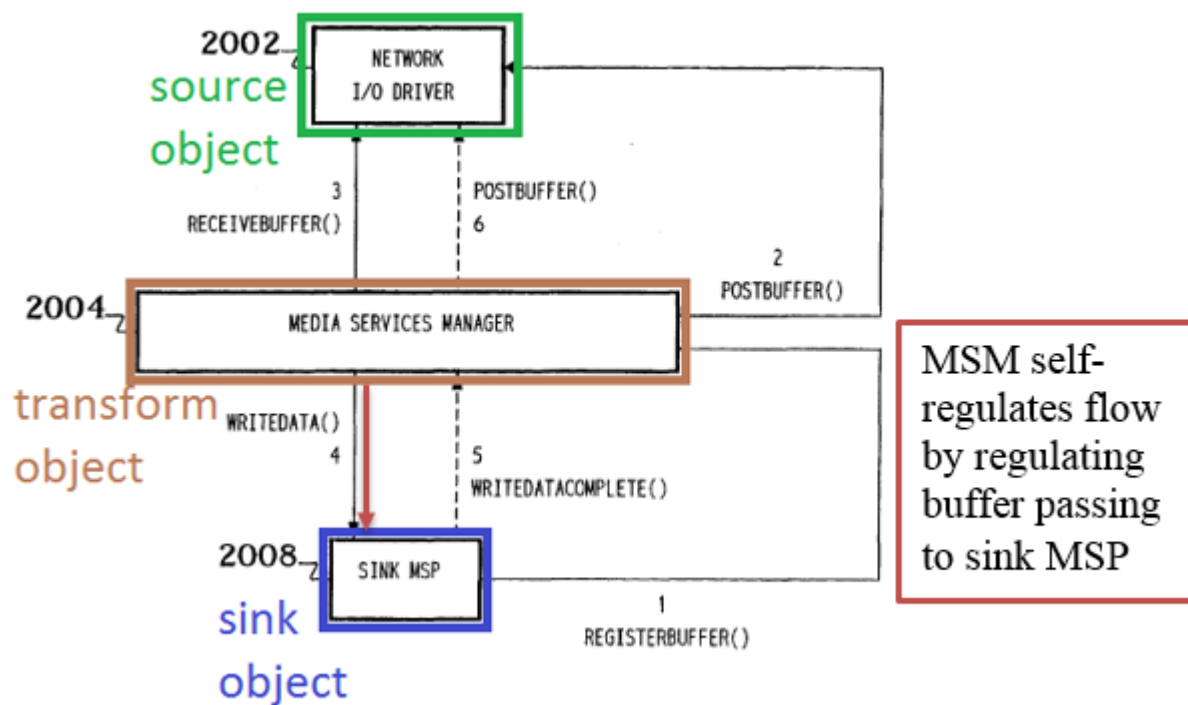
[31.l, 61.l] wherein said sink object is automatically flow controlled by said transform object;

Sampat discloses that the sink object is automatically flow controlled by the transform object. SE1003, ¶¶82-85, 90-94. Under BRI, “automatically flow controlled” includes “self-regulated” and, consequently, this claim element covers “wherein said sink object is *self-regulated* by said transform object.” SE1014, 101-106; SE1001, 8:52-65; SE1003, ¶83.

In Sampat, the MSM (transform object) self-regulates flow, including flow to the sink MSPs (sink object). SE1004, 14:43-54; SE1003, ¶¶84, 90-91. Sampat explains: “*MSM 1808 manages the flow* of data through the client software architecture” including “[f]rom the network to a sink media service provider

(MSP)” and “[f]rom mass storage device 1716 to a sink MSP.” SE1004, 14:43-54 (emphasis added).

The MSM regulates the flow of data to the sink object both by starting and stopping flow (SE1004, 15:36-16:57) and also by regulating buffer passing during flow (SE1004, 17:39-18:5). Sampat shows and describes its transform object (MSM) controlling flow in the same way as described in the ’389 patent, with the sink object (sink MSP) emptying buffers only when received from the MSM. SE1004 at 17:39-18:5; FIG. 20.



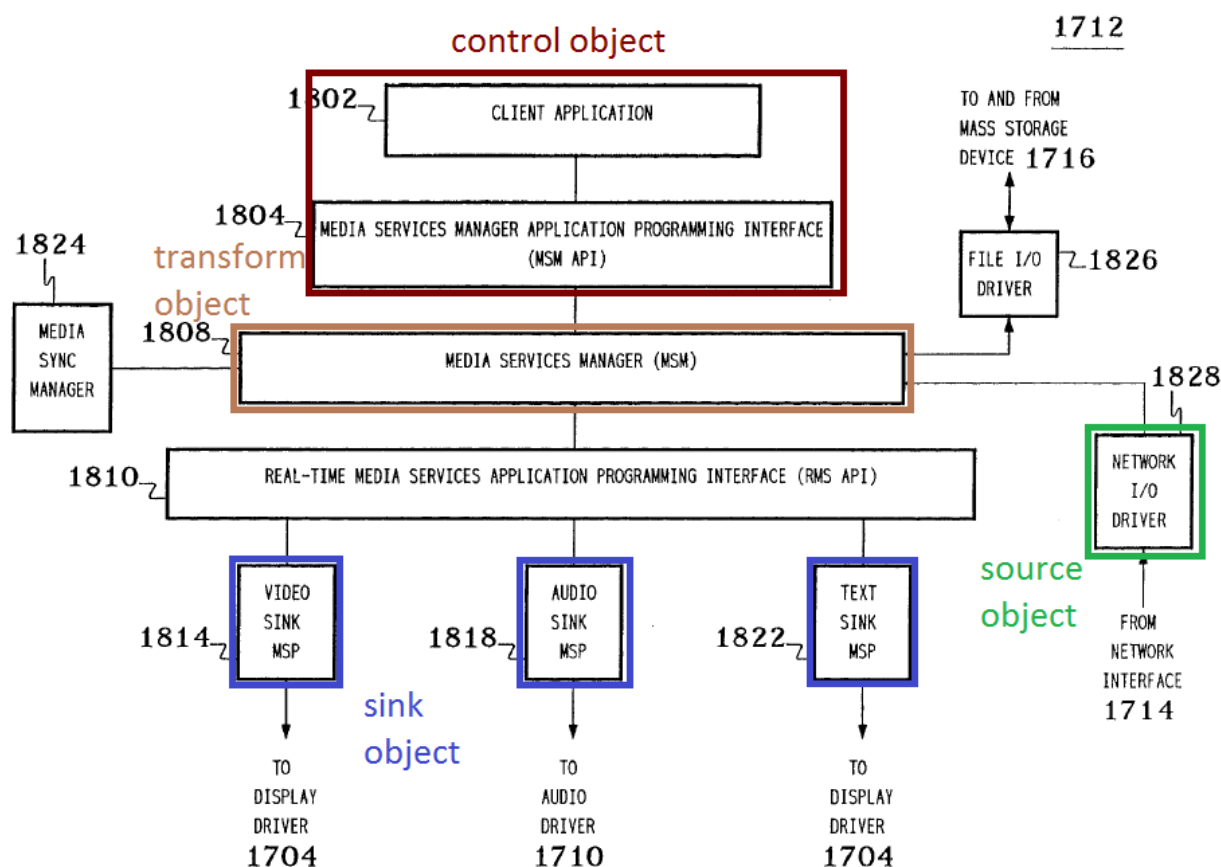
SE1004 at FIG. 20 (annotated), *see also* 17:59-63 (describing the MSM writing the sink buffer data to the sink MSP in step 4 before the sink MSP can play the sink buffer data in step 5); SE1003, ¶92. This description in Sampat is consistent with

the description of automatic flow control in the '389 patent, which equates waiting for buffers from the transform object to being “self-regulating” and “automatic flow control.” SE1001 at 8:45-51; *see also* 8:19-9:16; SE1003, ¶¶93. Accordingly, a POSITA would have understood that Sampat teaches its MSM (transform object) automatically controlling the flow of the sink object, not that the flow is merely reactive and independent of the MSM. SE1003, ¶¶94.

[31.m] providing [61.m] a control object,

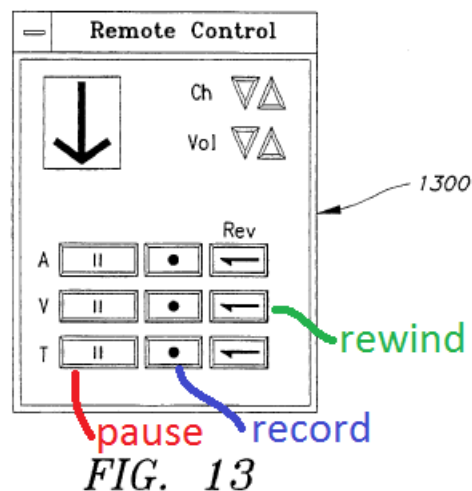
Sampat discloses providing a control object. SE1003, ¶¶98-106. Under BRI, the term “control object” includes “a collection of data and operations that receives commands from a user that control the flow of broadcast data.” SE1014, 92, 98. Sampat discloses a control object in the form of the client application 1802 and the media services manager (MSM) application programming interface (API) 1804. SE1004, 14:21-54, 15:4-6, FIG. 18. Client application 1802 and MSM API 1804 are a collection of data and operations. SE1004, 14:21-15:16; SE1003, ¶¶99. Figure 18, below, has the control object colored burgundy.

FIG. 18



SE1004, FIG. 18 (annotated). The client application 1802 (control object) uses the user interface shown in FIGS. 2-14 to control the flow of broadcast data through the system. SE1004, 14:34-27, 2:17-19, 39:51-54. “Remote control window functions include changing channels; changing audio volume; and playing, recording, or rewinding the audio, video, or text components of the current channel.” SE1004, 7:28-31. Therefore, a POSITA would have understood that Sampat’s client application/MSM API controls the flow of broadcast data through the system by playing, pausing, recording, and rewinding. SE1003, ¶100. Figure

13 shows buttons for receiving user commands for pausing, recording, and rewinding broadcast data.



SE1004, FIG. 13 (annotated), *see also* 7:28-31; 39:55-62 (“record, play, and rewind”). Accordingly, a POSITA would have understood that Sampat teaches the control object receiving commands that control the flow of broadcast data through the system from the user. SE1003, ¶¶101-101.

[31.n, 61.n] wherein said control object receives commands from a user, said commands control the flow of the broadcast data through the system; and

As described above at [31.m, 61.m], Sampat discloses this wherein clause, which is identical to language in the construction of control object as addressed. SE1003, ¶¶99-101.

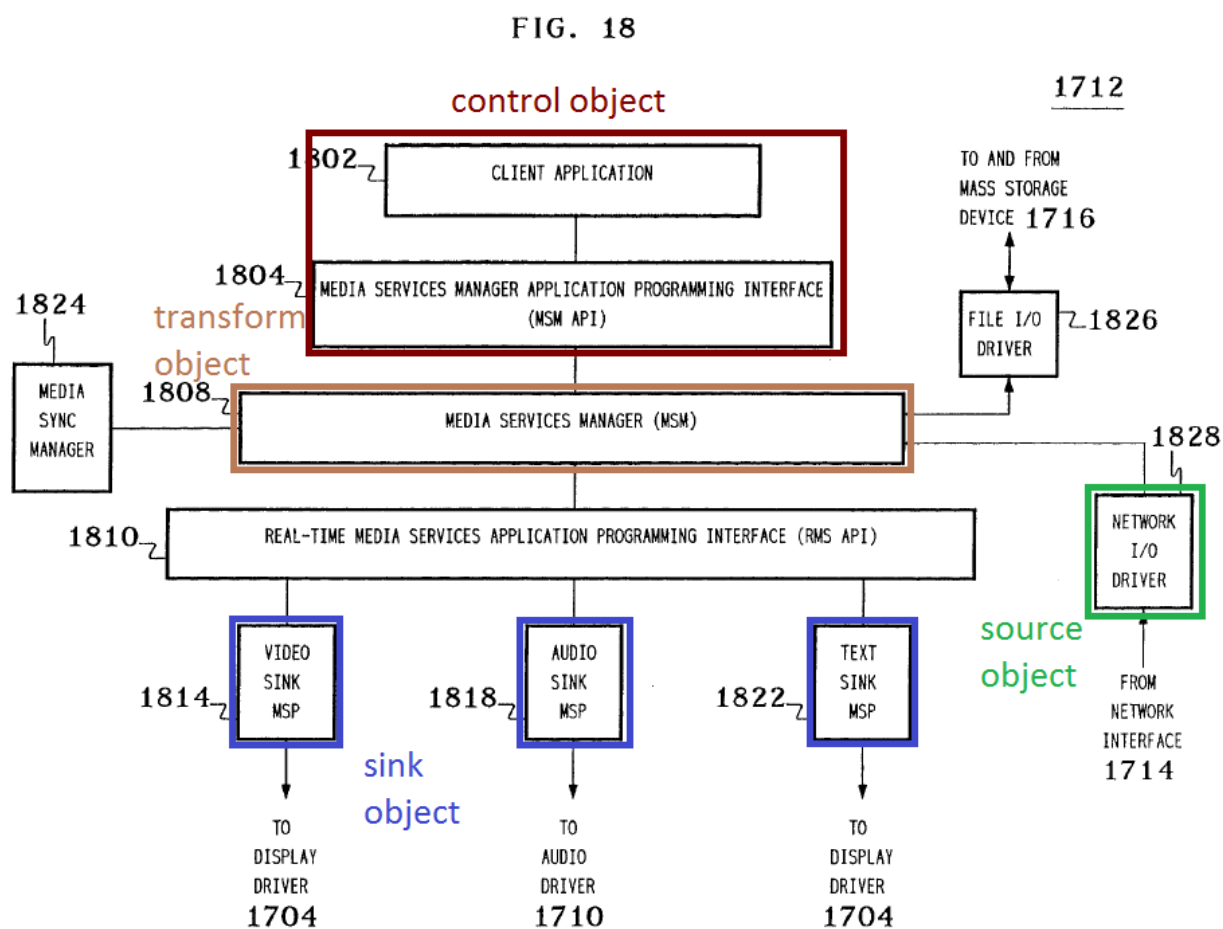
[31.o, 61.o] wherein said control object sends flow command events to said source, transform, and sink objects.

Sampat discloses the control object sending flow command events to the source, transform, and sink objects. SE1003, ¶¶102-106. Under BRI, this claim

element includes: “the control object sends information relating to a change of condition in the flow of the broadcast data to the source, transform and sink objects.” SE1013, 16-17.

Sampat explains that the control object (client application 1802 and MSM API 1804) sends flow command events to the transform object (MSM) and also to the source and sink objects via the transform object. SE1004, 15:36-16:57.

Sampat shows this architecture in FIG. 18.



SE1004, FIG. 18 (annotated); SE1003, ¶103.

Sampat describes a number of examples of the control object sending information regarding a change of condition, such as starting, pausing, and shutting down. SE1004, 15:36-16:57; SE1003, ¶ 104. In one example of starting play, the user first “asks the client application to start processing specified data streams” and then the “client application passes the MSM a list of the data streams to be started.” SE1004, 15:66-16:8. The MSM then relays information to the source and sink objects relating to the change of condition. SE1004, 16:9-29.

The BRI for this claim element does not require the flow command events to pass *directly* to each object, only that information is sent to each object from the control object. SE1013, 16-17. This can be direct or indirect. For example, the ’389 patent describes flow command events going first to one object and then being relayed to other objects. SE1001, 8:21-24 (“To pause the pipeline, for example, an event called ‘pause’ is sent to the first object in the pipeline. The event is ***relayed on to the next object*** and so on down the pipeline.”) (emphasis added). Accordingly, a POSITA would have understood that Sampat discloses the control object sending flow command events to the source, transform, and sink objects. SE1003, ¶105.

Therefore, Sampat (client-side) teaches each and every element of both of claims 31 and 61. SE1003, ¶¶39-106.

B. GROUND 2: Claims 31 and 61 are Obvious under § 103 over Sampat (client-side) in view of VfW and SoundBlaster

As described above in Section IV.A, *supra*, Sampat (client-side) discloses each and every element of claims 31 and 61. To the extent that Sampat is found not to disclose one or more claim elements, such claim elements would have been obvious in view of the predictable combination of Sampat, VfW, and SoundBlaster. SE1003, ¶¶107-119. This is particularly true for claim elements [31.i]-[31.k] and [61.i]-[61.k].

For example, to the extent that Sampat does not *explicitly* say that the disclosed display driver includes a video decoder (*see* claim elements [31.j], [61.j]), such decoders were commonly known in such drivers. SE1003, ¶108. VfW confirms that the Microsoft Video for Windows display driver includes a video decoder. SE1007, 8 (“video compression and decompression drivers”), 157-211 (video decompression); *see also* SE1008, 2:10-40; SE1021, 2. This is the same “Video for Windows” display driver that Sampat describes as the preferred embodiment. SE1004, 14:10-12 (“preferably Microsoft Video for Windows.”). A POSITA would have understood that Sampat’s disclosure of outputting streams to a display driver such as Microsoft Video for Windows would output the streams to a video decoder. SE1003, ¶108.

Multiple reasons would have prompted a POSITA to include a video decoder in the display driver as suggested by Video for Windows in the system of

Sampat.

First, a POSITA would have been prompted to implement a video decoder in the display driver of Sampat because Sampat uses encoded data (e.g., MPEG) and video displays required encoded data to be decoded. SE1003, ¶110. Indeed, Sampat describes the display driver 1704 as being used “for processing” prior to “display on monitor 1708.” SE1004, 13:55-57. A POSITA would have recognized that any coded data would require decoding in order for the display to properly reproduce a video stream, and would have recognized that the display driver should include a decoder as was traditional in such systems. SE1003, ¶110.

Second, a POSITA would have been prompted to implement a video decoder in the display driver of Sampat because Sampat specifically identifies “Microsoft Video for Windows” as the preferred display driver. SE1004, 14:10-12. Video for Window included decoders. SE1007, 8, 157-211. This disclosure in Sampat that Video for Windows is preferred would have motivated a POSITA to use that very driver. SE1003, ¶111.

Third, a POSITA would have been prompted to implement a video decoder in the display driver of Sampat because doing so would be merely the application of a known technique (e.g., decoding video using a display driver) to a known system (e.g., Sampat’s system having a display driver) ready for improvement to yield predictable results. SE1003, ¶112. Indeed, “when a patent ‘simply arranges

old elements with each performing the same function it had been known to perform' and yields no more than one would expect from such an arrangement, the combination is obvious." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 417 (2007). Here, both Sampat and VfW disclose the same display driver, and a POSITA would have recognized that applying the suggestions in VfW to Sampat's system would have led to predictable results without significantly altering or hindering the functions performed by the system of Sampat. SE1003, ¶112.

Additionally, to the extent that Sampat does not *explicitly* say that the sink outputs to an audio decoder (*see* claim elements [31.j] and [61.j]), such decoders were commonly known. SE1003, ¶113. For example, SoundBlaster discloses decoding in the form of "decompression." SE1006, 152; SE1003, ¶113. In particular, SoundBlaster discloses "ADPCM, hardware decompression" associated with various compression schemes. SE1006, 152. A POSITA would have understood that ADPCM decompression stands for "adaptive differential pulse-code modulation," which is used for coding and decoding an audio signal. SE1003, ¶113; SE1006, 152. SoundBlaster also discloses having a "Digital to Analog Converter," which a POSITA would have understood performs decoding. SE1006, 146, 152-153, 10; SE1003, ¶113. This SoundBlaster reference is the same "SoundBlaster" that Sampat describes as the preferred embodiment. SE1004, 4:56-57; 15:1-3; SE1003, ¶113.

Multiple reasons would have prompted a POSITA to implement outputting to an audio decoder (as suggested by SoundBlaster) in the system of Sampat.

First, a POSITA would have been prompted to implement the well-known prior art functionality of outputting to an audio decoder (as suggested by SoundBlaster) in the system of Sampat because Sampat uses encoded data (e.g., MPEG) and audio hardware (e.g., an amplifier and speakers) requires encoded data to be decoded. SE1003, ¶115. A POSITA would have understood that Sampat's system would be unable to process encoded audio data without a decoder, and SoundBlaster discloses a suitable and well-known option for providing such decoding. SE1003, ¶115; SE1006, 10, 146, 152-153.

Second, a POSITA would have been prompted to implement the well-known prior art functionality of outputting to an audio decoder (as suggested by SoundBlaster) in the system of Sampat because Sampat specifically identifies "SoundBlaster Pro from Creative Labs" as the preferred device for enabling sound. SE1004, 4:56-57. SoundBlaster included decoders. SE1006, 10, 146, 152-153. Sampat's disclosure of preferring SoundBlaster would have motivated a POSITA to use that very device and its decoder, which would have been conventional, well-known, and predictable. SE1003, ¶116.

Third, a POSITA would have been prompted to implement the well-known prior art functionality of outputting to an audio decoder (as suggested by

SoundBlaster) in the system of Sampat because doing so would be merely the application of a known technique (e.g., decoding audio using a decoder) to a known system (e.g., Sampat's system including sound) ready for improvement to yield predictable results. SE1003, ¶117; *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. at 417. Here, Sampat and SoundBlaster disclose the same SoundBlaster device and a POSITA would have recognized that applying the suggestions in SoundBlaster to Sampat's system would have led to predictable results without significantly altering or hindering the functions performed by the system of Sampat. SE1003, ¶117.

Accordingly, a POSITA would have understood that Sampat in view of VfW and SoundBlaster renders obvious providing a sink object that obtains data stream buffers from the transform object and outputs the streams to a video and audio decoder—thus meeting claim elements [31.i]-[31.j] and [61.i]-[61.j]. SE1003, ¶118.

As explained above in [31.k] and [61.k], Sampat describes the decoder (display driver 1704) converting streams into display signals and sending the signals to a display. SE1003, ¶119. VfW also discloses its decoder converting streams into display signals and sending those signals to a display monitor. SE1007, 157-158 (converting compressed video data streams and sending to a “Video Display”), 158-159 (“a decompression driver with the ability to write to the

video display”). Accordingly, claim elements [31.k] and [61.k] are also obvious over the combination of Sampat, VfW, and SoundBlaster.

C. GROUND 3: Claims 31 and 61 are Anticipated under § 102(b) by Sampat (server-side)

Claims 31 and 61, and all of their claim elements, are taught by Sampat. As explained above, Sampat teaches the claim elements when viewed from both the server 102 and the client 104, and Ground 3 addresses Sampat when viewed from Sampat’s server 102 (not the client 104, as in Grounds 1-2). For Ground 3, FIG. 1 has been annotated below to schematically show that the input devices are upstream of the server 102, and that the rest of the claim elements are in the server 102.

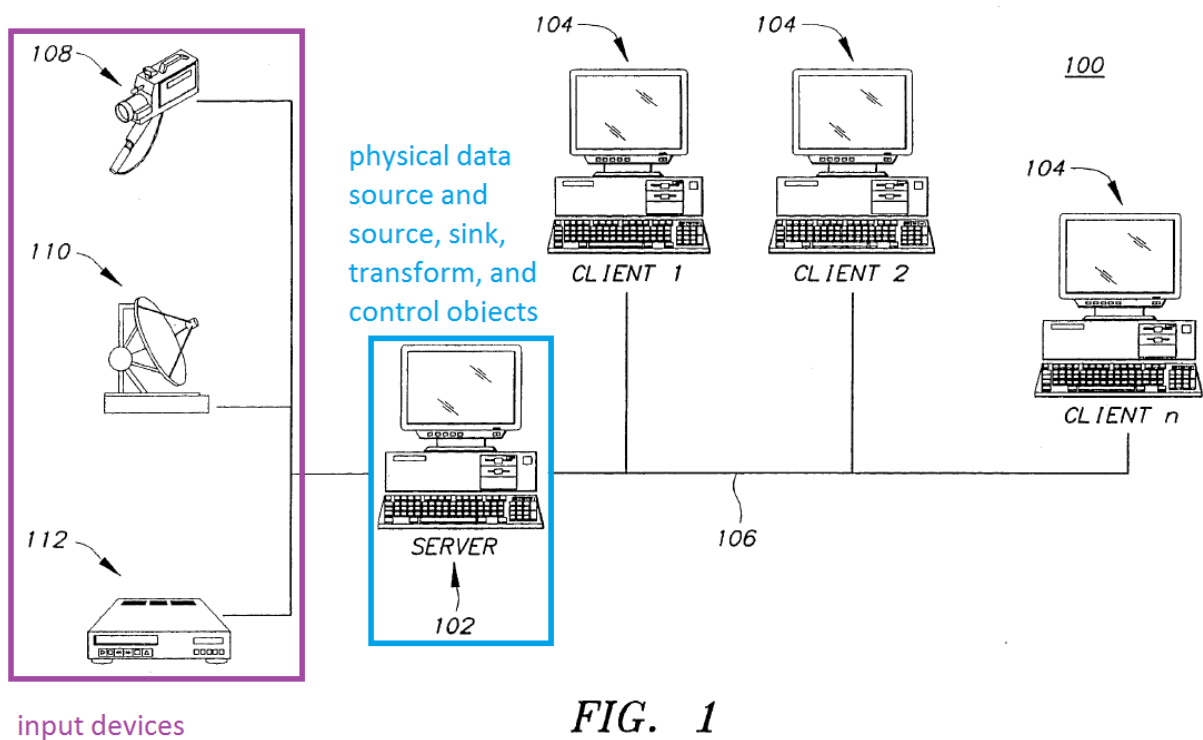
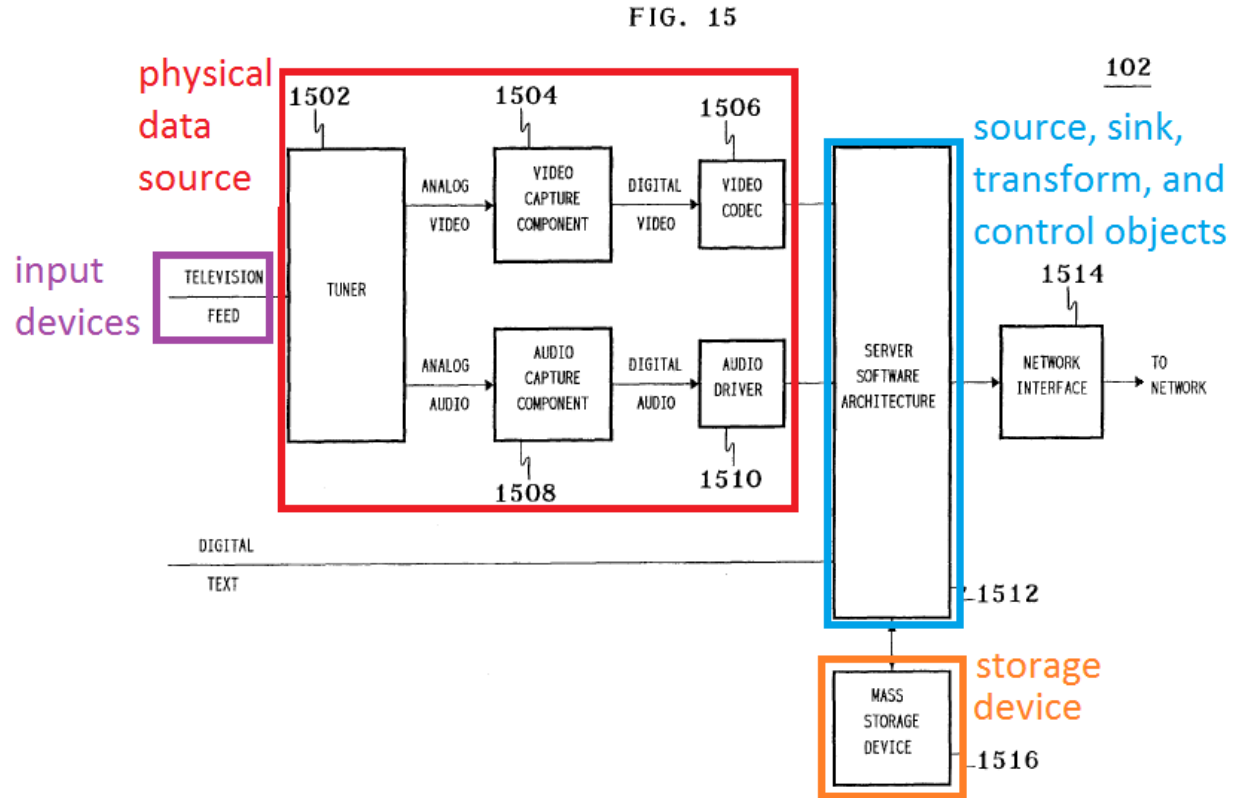


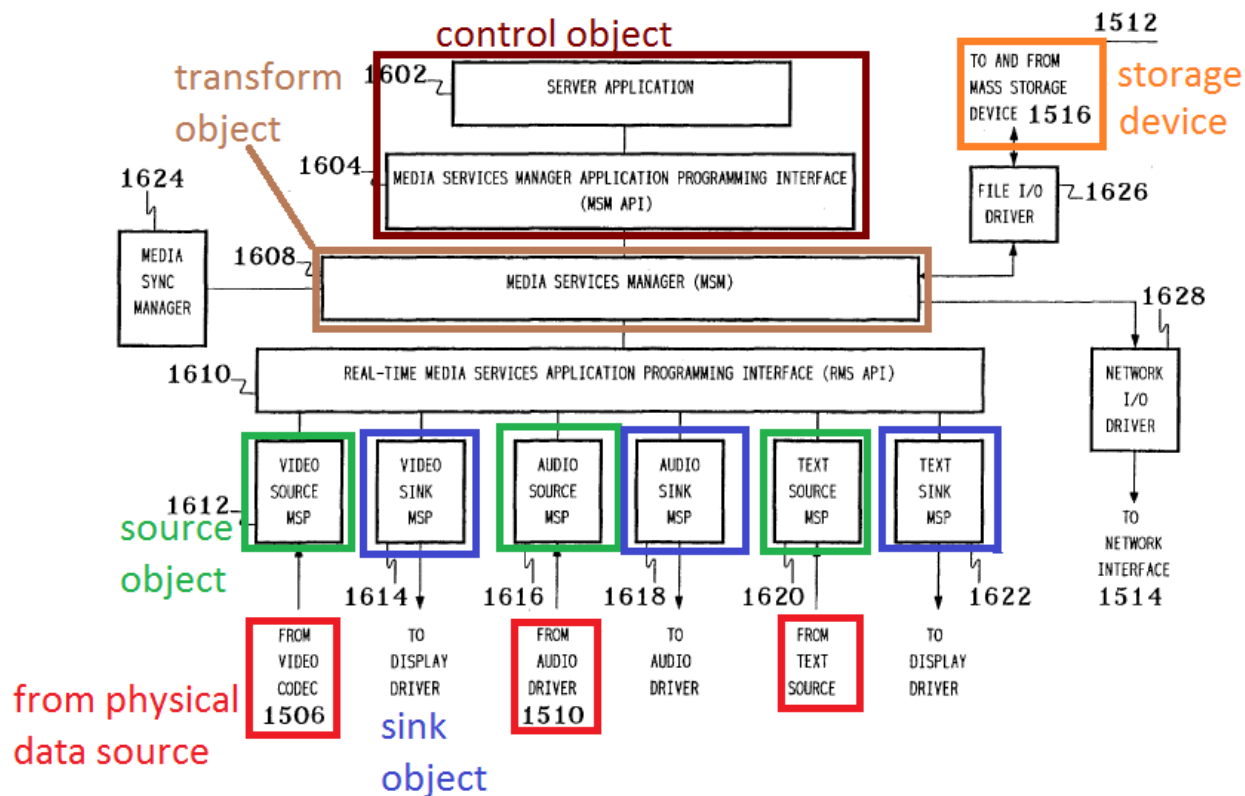
FIG. 1

SE1004, FIG. 1 (annotated for Grounds 3 and 4); SE1003, ¶120. Figures 15 and 16 show the source, transform, sink, and control objects within the server 102.



SE1004, FIG. 15 (annotated).

FIG. 16



SE1004, FIG. 16 (annotated); SE1003, ¶121. These features will be discussed with respect to the claim elements, which will be considered in order for each of claims 31 and 61.

[31.Pre] A process for the simultaneous storage and play back of multimedia data, comprising the steps of:

[61.Pre] An apparatus for the simultaneous storage and play back of multimedia data, comprising:

As explained above at [31.Pre, 61.Pre] in Ground 1 (Section IV.A, *supra*), the preamble of claims 31 and 61 are not limiting. SE1014, 82. Even if the preamble were limiting, Sampat discloses an apparatus (which in Ground 3 is the

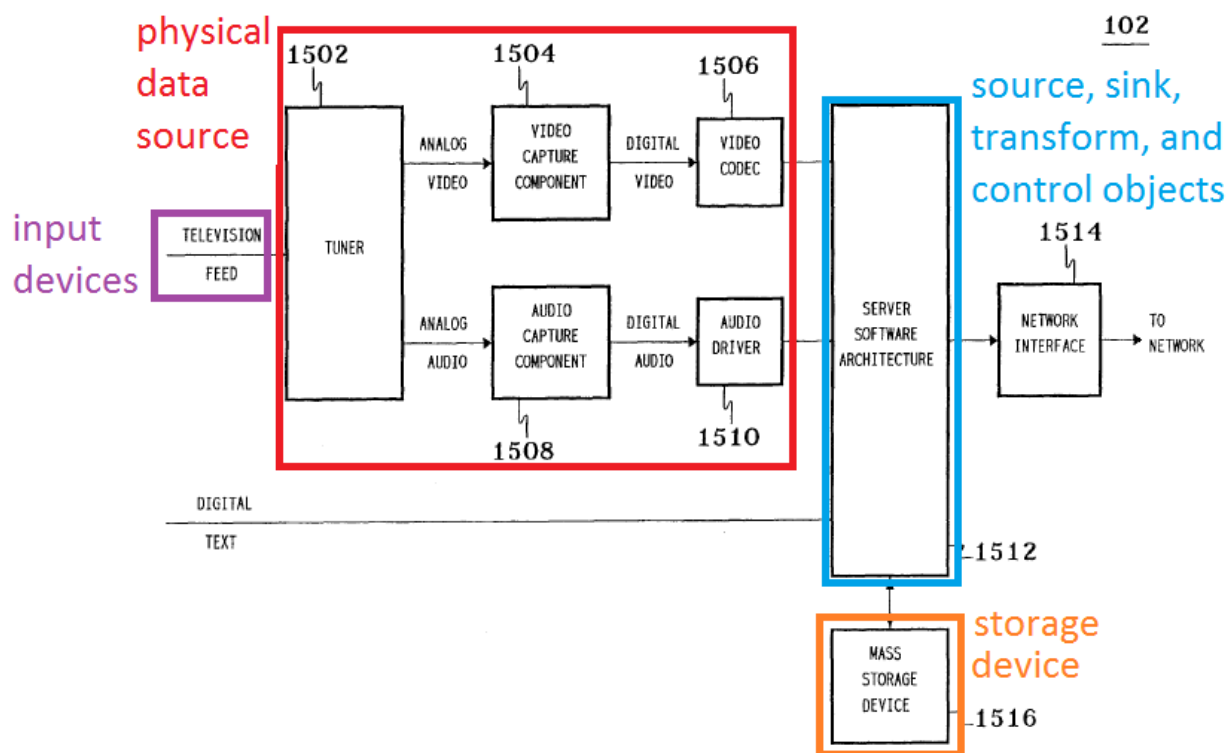
server 102) and corresponding process for the simultaneous storage and play back of multimedia data. SE1004, 8:18-22, 13:59-63, 2:10-16; SE1003, ¶125.

[31.a] providing [61.a] a physical data source,

Sampat discloses providing a physical data source. SE1003, ¶¶126-136. The BRI of “physical data source” includes “hardware and software that parses video and audio data from said broadcast data.” SE1014, 109-111; SE1001, 8:43-45. The BRI of “parses video and audio data from said broadcast data” includes “analyzes video and audio data from the broadcast data.” SE1014, 82-88; SE1001, 5:3-9, 6:37-46.

As viewed from the server-side, Sampat discloses a physical data source (hardware and software) in the form of the tuner 1502 in conjunction with the video capture component 1504, video codec 1506, audio capture component 1508, and audio driver 1510. SE1004, 7:59-8:11. Figure 15, reproduced below, shows the physical data source (red).

FIG. 15



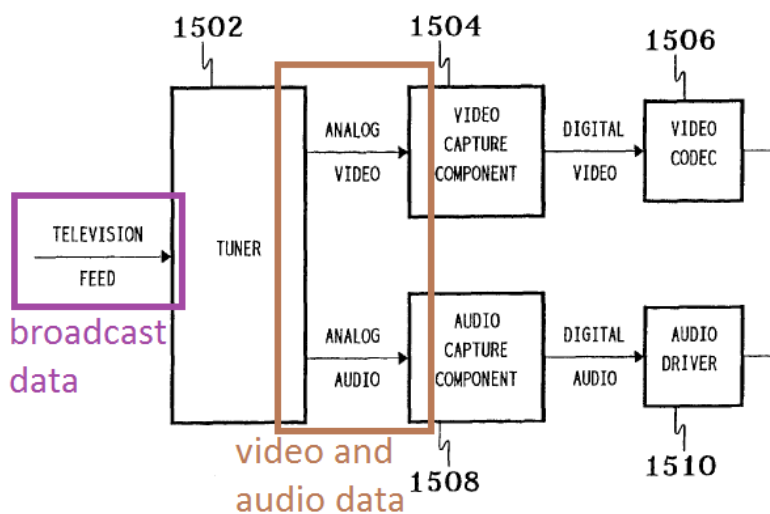
SE1004, FIG. 15 (annotated); SE1003, ¶128.

Sampat discloses physical data source (the tuner 1502, etc.) accepting broadcast data from one or more input devices (e.g., camera 108, antenna 110, and VCR 112). SE1004, 4:7-14 (“Server 102 is capable of capturing analog audio and video signals from three different sources: (1) signals generated locally by camera 108, (2) signals received by antenna 110 from a remote source, and (3) recorded signals from VCR 112”), 38:42-54 (the system can be used with “broadcasting”). “For example, server 102 may receive via antenna 110 a first television program” and “may receive a second television program ... from VCR 112.” *Id.*, 4:15-23; *see also* SE1001, 3:32-43 (describing receiving broadcast data from a “TV source

transmitter”), 11:50-51, FIG. 13 (describing and showing a VCR as another input device); SE1003, ¶129.

Sampat further discloses the physical data source parsing video and audio data from the broadcast data. SE1004, 7:55-65, 4:24-43. “In particular, tuner 1502 of server subsystem 102 receives, demodulates, and splits one or more analog television feed signals into their constituent analog audio and video signals.” *Id.* Figure 15, reproduced below, shows a block diagram of the tuner 1502 parsing the broadcast data (purple) into video and audio data (brown).

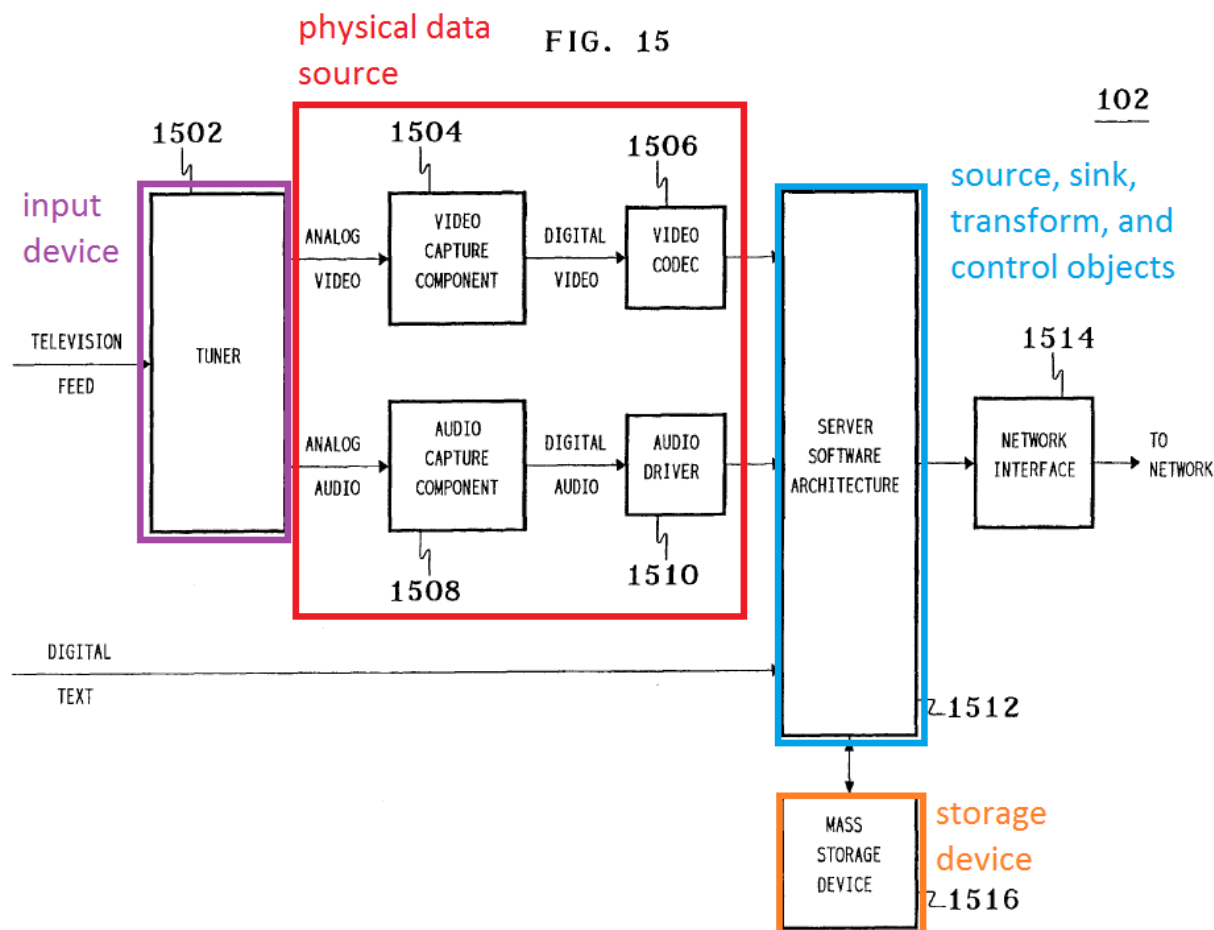
FIG. 15



SE1004, FIG. 15 (annotated); SE1003, ¶¶130-131.

Alternatively, Sampat discloses a physical data source (hardware and software) in the form of the combination of the video capture component 1504, video codec 1506, audio capture component 1508, and audio driver 1510, *without*

the tuner 1502. SE1003, ¶134. In that case, the tuner 1502 is the input device, as shown below.



SE1004, FIG. 15 (annotated-alternative).

Such alternative interpretations are consistent with Patent Owner's infringement contentions. *See* SE1015, 39-48, 101-110. For example, Patent Owner alleges that the "Samsung DVR has an input device that may include cable input, tuners..." (SE1015, 39), which is consistent with Sampat's disclosure of tuner 1502 (SE1004, 7:59-61, 8:22-25); SE1003 ¶135.

Sampat also discloses that this physical data source *parses* video and audio data under the BRI, which includes “*analyzes* video and audio data from said broadcast data.” SE1014, 82-88. The physical data source analyzes video and audio data from the broadcast data in multiple ways. For example, “[v]ideo capture component 1504 captures and converts the analog video signals into digital video data streams” and “audio capture component 1508 captures and converts the analog audio signals into digital audio data streams,” which includes analysis of video and audio data when converting into digital video. SE1004, 7:62-65; SE1003, ¶136. Additionally, “[v]ideo codec 1506 compresses the digital video data streams” and “[a]udio driver 1510 places the audio data into buffers,” which also includes analysis of video and audio data as broadly construed. SE1004, 8:7-10; SE1003, ¶136. Accordingly, Sampat also discloses the claimed physical data source under this alternative interpretation. SE1003, ¶¶134-136.

[31.b, 61.b] wherein said physical data source accepts broadcast data from an input device, parses video and audio data from said broadcast data, and temporarily stores said video and audio data;

Sampat discloses that the physical data source accepts broadcast data from an input device, parses video and audio data from said broadcast data, and temporarily stores said video and audio data. SE1003, ¶¶126-136. As explained in claim element [31.a, 61.a], Sampat discloses that the physical data source accepts broadcast data from an input device and parses video and audio data from said

broadcast data. SE1003, ¶¶126-131, 134-136; SE1004, 4:7-14, 4:24-43, 7:55-8:11, 38:42-54, FIG. 15.

Sampat further discloses the physical data source temporarily storing the video and audio data. SE1003, ¶132. For example, Sampat discloses that the part of the physical data source that includes the video capture component 1504 and codec 1506 is hardware such as “an Intel® SmartVideo® Recorder (ISVR)” and that the part of the physical data source that includes the audio capture component 1508 is “a Creative Labs SoundBlaster Pro.” SE1004, 8:25-31. A POSITA would have recognized that the ISVR and the Creative Labs SoundBlaster Pro would temporarily store the video and audio data as that data is being processed. SE1003, ¶132. Such ISVR and SoundBlaster Pro devices were well-known to a POSITA prior to the 1998 filing date of the ’389 patent, and were also known to include temporary storage of video and audio data. SE1003, ¶132; *see, e.g.*, SE1006, 20, 92, 98, 146, 153, 158; SE1005, 5:10-16, 5:20-60; FIG. 3.⁴

Indeed, a POSITA would have understood Sampat’s disclosure of “capturing” the video and audio as meaning that the components *store* the video

⁴ The exhibits SE1005 and SE1006 are offered in Ground 3 only to show what was well-known about the ISVR and SoundBlaster Pro devices—not that these references should be combined to show obviousness in Ground 3.

and audio data. SE1004, 7:62-65, 31:65-32:3 (“video *capture* component 1504 and audio *capture* component 1508 of server 102 of FIG. 15 may *capture* data at different rates”) (emphasis added). In fact, the term “capture” as used in this context conventionally meant that such data was stored. SE1009, 2 (“cause (data) to be stored in a computer”); SE1010, 2 (“The recording of data...into a computer.”); SE1003, ¶133. Accordingly, Sampat discloses that the physical data source accepts broadcast data from an input device, parses video and audio data from said broadcast data, and temporarily stores said video and audio data. SE1003, ¶¶126-136.

[31.c] providing [61.c] a source object,

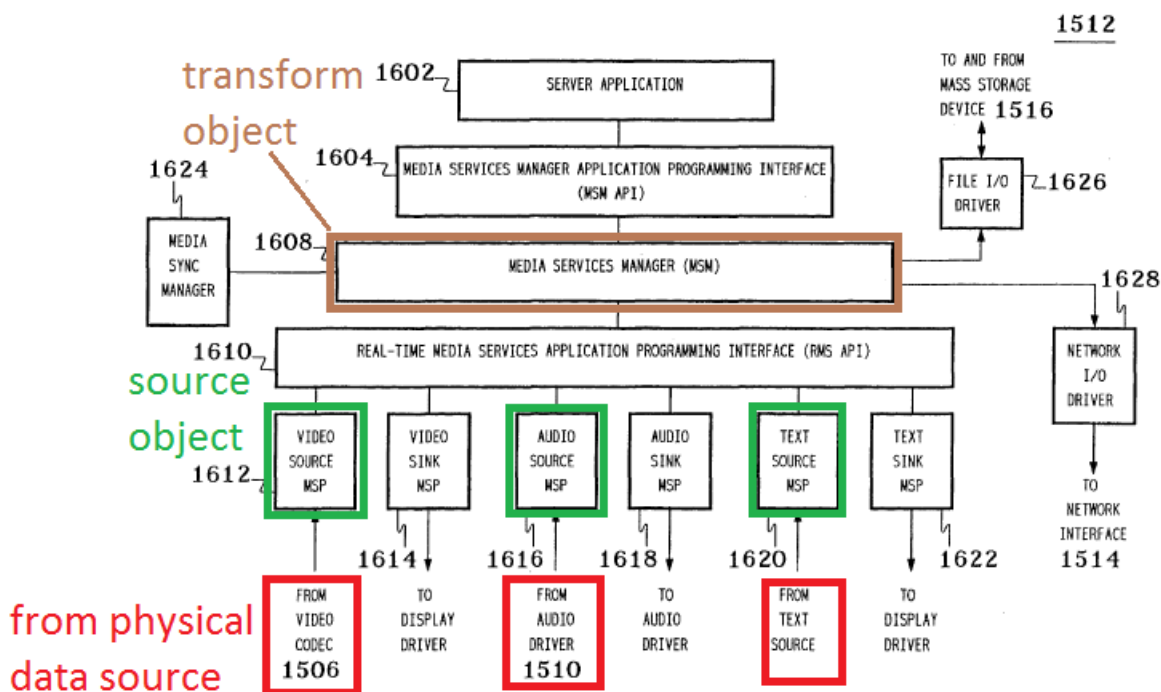
Sampat discloses providing a source object. SE1003, ¶¶137-145. The BRI for “source object” includes “a collection of data and operations that (1) extracts video and audio data from a physical data source, (2) obtains a buffer [memory where data can be temporarily stored for transfer] from a transform object, (3) converts video data into data streams, and (4) fills the buffer [memory where data can be temporarily stored for transfer] with the streams.” SE1014, 91-98; SE1001, 8:39-51.

As viewed from the server-side, Sampat describes a source object in the form of one or more source MSPs: “A source MSP is a media service provider that assists in the receipt of a data stream from an external source or local mass storage

device.” SE1004, 9:37-40. The source MSPs are a collection of data and operations. SE1004, 9:7-56. The source MSPs also perform each of the four functions defined as being performed by a source object. SE1003, ¶139.

First, the source MSPs extract audio and video from the physical data source (e.g., from the video codec 1616 and audio driver 1510). SE1004, 9:46-51. (“Video source MSP 1612 receives a video data stream from video codec 1506...audio source MSP 1616 and text source MSP 1620 receive audio and text data streams from audio driver 1510.”). Figure 16 is reproduced showing the source object (the source MSPs, colored green) extracting video and audio data from the physical data source (colored red).

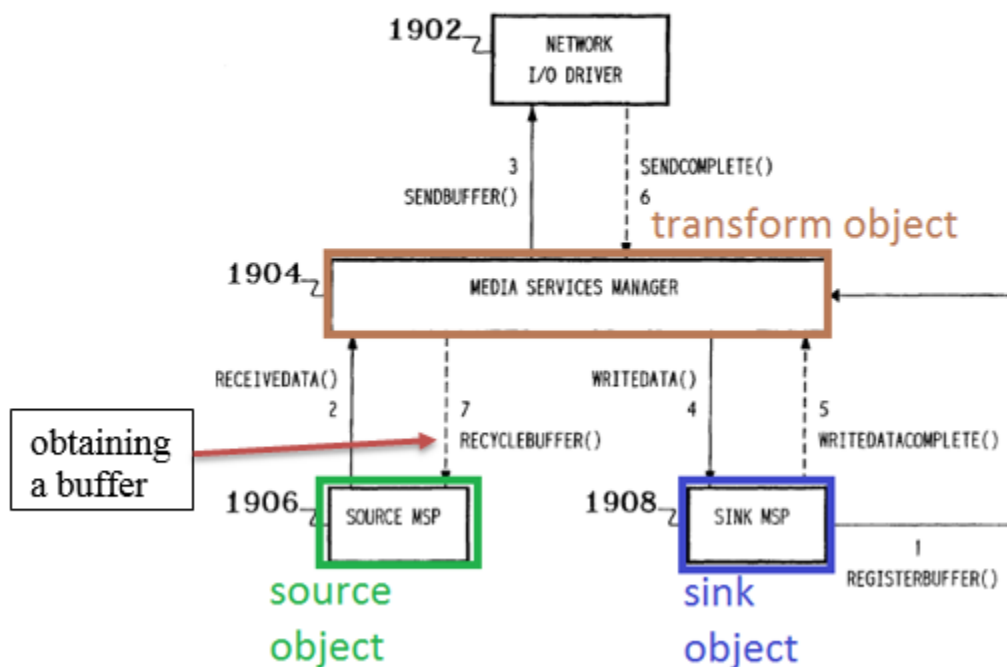
FIG. 16



SE1004, FIG. 16 (annotated); SE1003, ¶140.

Second, Sampat describes the source object (source MSPs) obtaining a buffer from the transform object (MSM), stating: “MSM recycles the buffer to the appropriate source MSP.” SE1004, 12:64-13:3. This is also shown in Figure 19, below.

FIG. 19



SE1004, FIG. 19 (annotated); SE1003, ¶141.

Third, Sampat discloses that the source object (source MSP) converts video data into time-stamped data streams “as the source MSPs stamp their data with the appropriate capture time.” SE1004, 11:46-55, SE1003, ¶¶142-143.

In the server, when a source MSP (1612, 1616, or 1620 of FIG. 16) receives new data, the MSP asks MSM 1608 for a new time-stamp from media sync manager 1624, which the MSP adds to the data header before sending the data to MSM 1608 for transmission to the network and/or storage to mass storage device 1516.

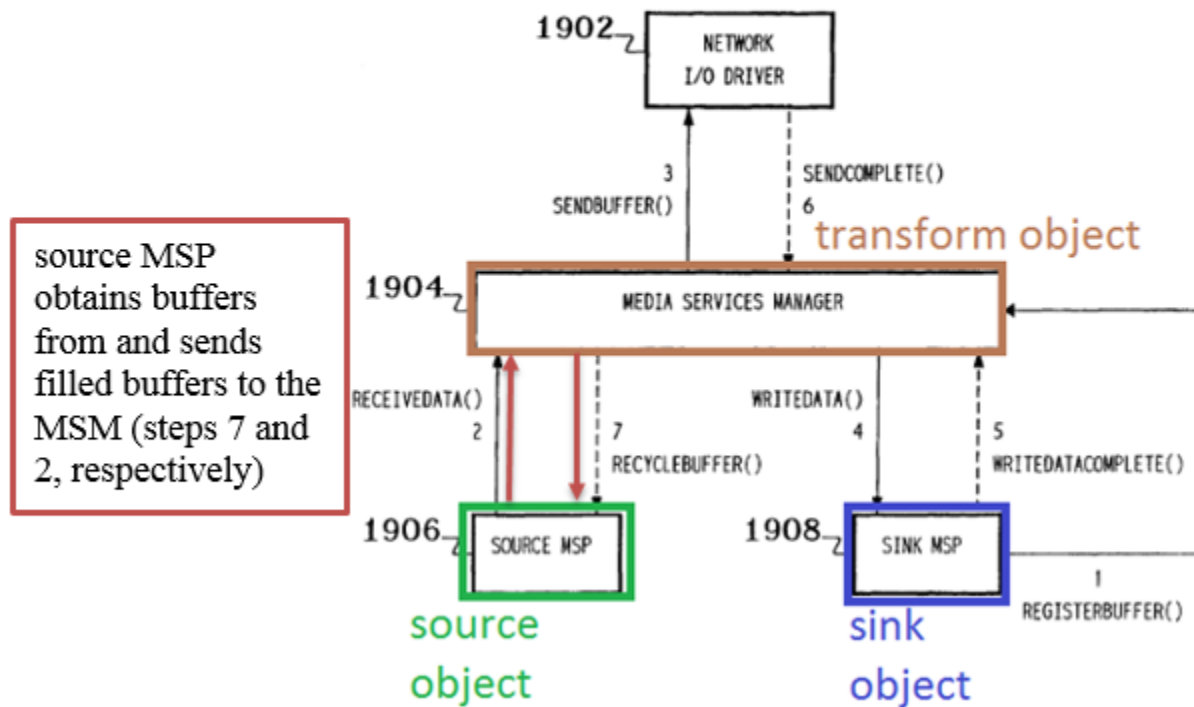
SE1004, 32:6-18.

Fourth, Sampat discloses that the source object (source MSPs) fills the buffer with data streams: “The source MSP is then free to refill the buffer.”

SE1004, 13:2-4, 17:16-19 (“fills them with data”). Figure 19 shows the source MSP obtaining a buffer from the MSM and sending the filled buffer back to the

MSM.

FIG. 19



SE1004, FIG. 19 (annotated); SE1003, ¶144.

[31.d, 61.d] wherein said source object extracts video and audio data from said physical data source;

As described above at [31.c, 61.c], Sampat discloses this wherein clause, which is identical to language in the construction of source object as addressed.

SE1003, ¶140.

[31.e] providing [61.e] a transform object,

Sampat discloses providing a transform object. SE1003, ¶¶146-153. Under BRI, a “transform object” is “a collection of data and operations that transforms the form of data upon which it operates.” SE1014, 101-106; SE1001, 7:48-8:65. The ’389 patent describe one type of transform as a *temporal* transform. SE1001, 7:64-8:8 (describing a transform as “temporal” when it “writes the buffer to a file 804 on the storage medium” and then pulls the buffer out “at a later time”).

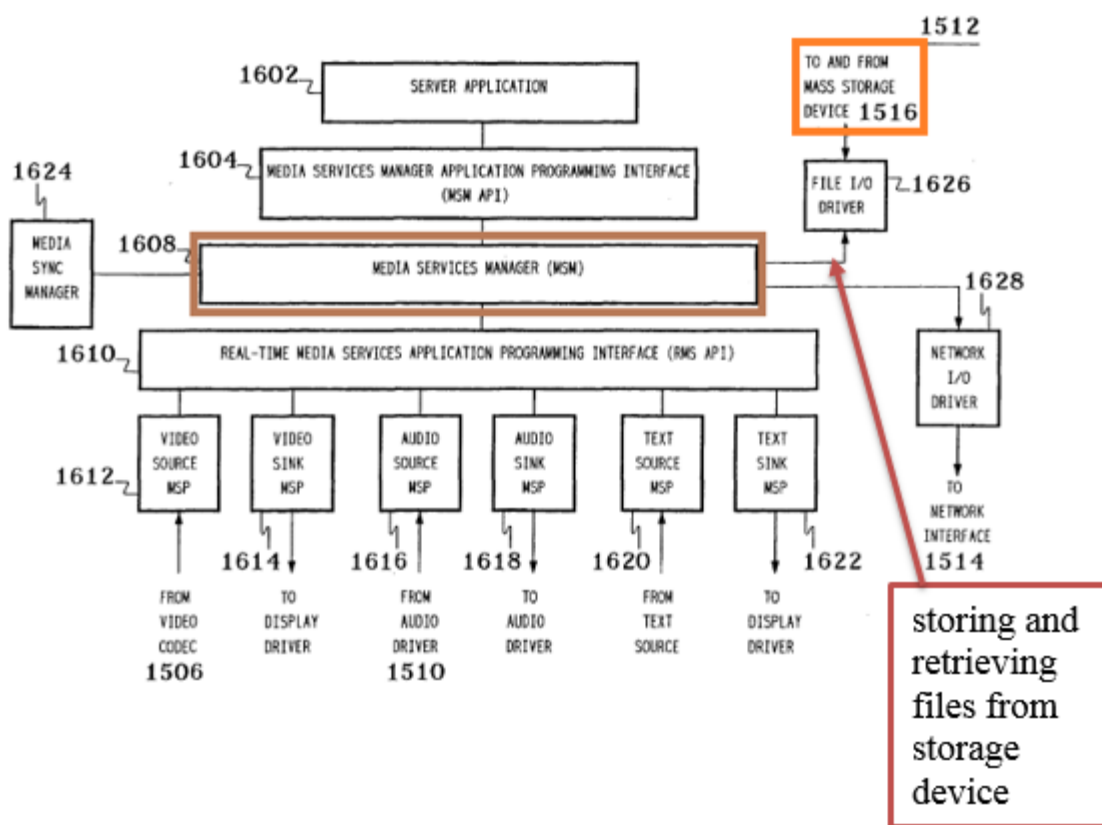
Sampat describes a transform object in the form of the “MSM” (MSM 1608 in Figure 16 and MSM 1904 in Figure 19). SE1004, 9:10-12, 17:8-38, FIGS. 16 and 19. The MSM is a collection of data and operations. SE1004, 8:50-9:45, FIGS. 16, 19; SE1003, ¶148. Sampat describes the MSM transforming the form of data upon which it operates by performing a temporal transform, which, the MSM performs by writing data to the mass storage device 1516 and retrieving the data at a later time for playback. SE1004, 10:10-11 (“MSM 1608 uses file I/O driver 1626 to store and retrieve data to and from mass storage device 1516.”), 8:17-21; FIGS. 15-16; SE1003, ¶149. Sampat describes the MSM temporarily storing and retrieving data flowing:

through MSM1608...From a source MSP to mass storage device 1516 (for storage of data received from an external source for subsequent processing), From mass storage device 1516 to the network (for multicasting of locally recorded data), and From mass storage device

1516 to a local sink MSP (for monitoring the processing of locally recorded data).

SE1004, 9:10-26. The MSM 1608 is shown transferring data streams “to and from mass storage device 1516” (quoted text colored orange) in FIG. 16 via the file I/O driver 1626. SE1004, FIG. 16, *see also* FIG. 15.

FIG. 16



SE1004, FIG. 16 (annotated); SE1003, ¶150.

The temporal transform performed by Sampat’s MSM is consistent with the ’389 patent’s temporal transform. *See* 1001 at 7:64-8:8 (describing a transform as “temporal” when it “writes the buffer to a file 804 on the storage medium” and

then pulls the buffer out “at a later time”). Thus, Sampat discloses a transform object that transforms the form of data upon which it operates in a similar manner to the ’389 patent. SE1003, ¶¶151-153.

[31.f, 61.f] wherein said transform object stores and retrieves data streams onto a storage device;

As described above at [31.e, 61.e], Sampat discloses this wherein clause, which is identical to language in the construction of transform object as addressed. SE1003, ¶¶149-153.

[31.g, 61.g] wherein said source object obtains a buffer from said transform object, said source object converts video data into data streams and fills said buffer with said streams;

As described above at [31.c, 61.c], Sampat discloses this wherein clause, which is identical to language in the construction of source object as addressed. SE1003, ¶141.

[31.h, 61.h] wherein said source object is automatically flow controlled by said transform object;

Sampat discloses that the source object is automatically flow controlled by said transform object. SE1003, ¶¶166-174. Under BRI, “automatically flow controlled” includes “self-regulated” and, consequently, this claim element includes “wherein said source object is ***self-regulated*** by said transform object.” SE1014, 101, 106 (emphasis added). District courts have repeatedly reached this construction and agreed that the analysis in the second reexamination of the ’389

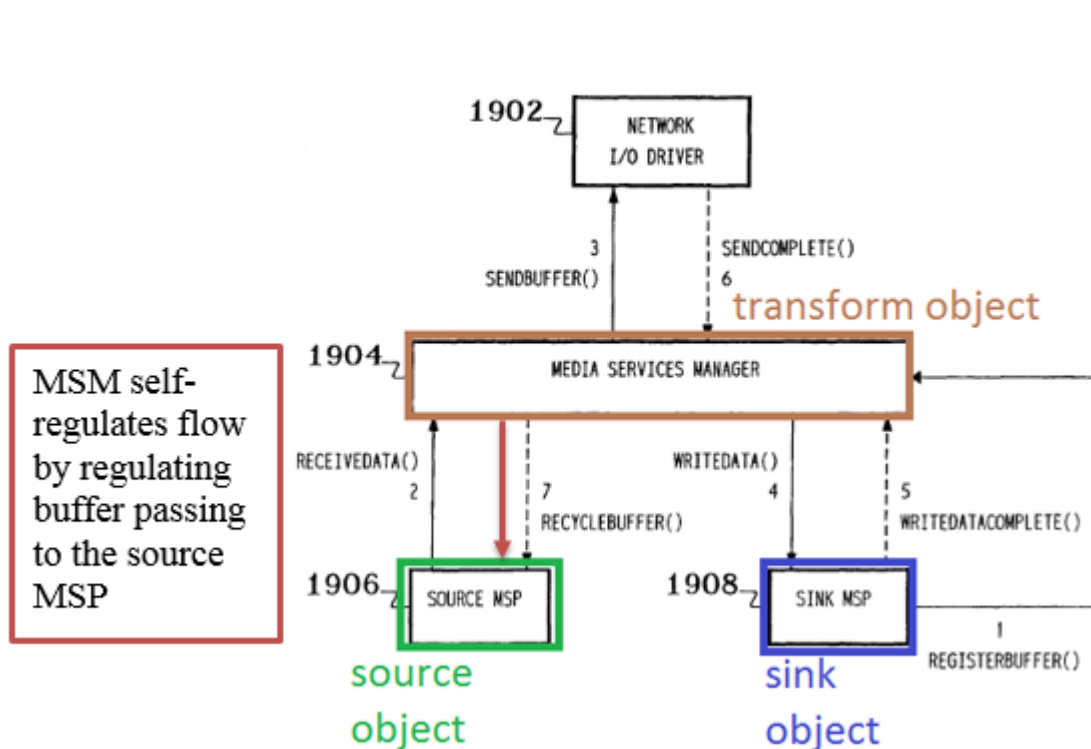
patent “did not limit or redefine” this relatively broad construction of “automatically flow controlled.” SE1014, 106; SE1013, 12; SE1003, ¶167.

In Sampat, the MSM (transform object) self-regulates flow, including flow from the source MSPs (source object). SE1004, 14:43-54; SE1003, ¶168. Sampat explains: “Media services manager (MSM) 1608 *manages the flow of data* through server software architecture 1512” including “[f]rom a source MSP to a local sink MSP” and “[f]rom a source MSP to mass storage device 1516.” SE1004, 9:10-26.

The MSM regulates the flow of data from the source object both by starting and stopping flow (SE1004, 9:57-10:7, *see also* 10:26-29) and also by regulating buffers passing during flow (SE1004, 16:60-17:39). Sampat shows and describes its transform object (MSM) controlling flow in the same way as described in the ’389 patent, with the source object (source MSP) filling buffers only when

received from the MSM. SE1004, 16:60-17:7; FIG. 19.

FIG. 19



SE1004, FIG. 19 (annotated); *see also* 17:35-38 (describing recycling the buffer from the MSM to the source MSP in step 7); SE1003, ¶¶170-172. This description in Sampat is consistent with the description of automatic flow control in the '389 patent, which equates waiting for buffers from the transform object to being “self-regulating” and “automatic flow control.” SE1001, 8:45-51; *see also* 8:19-9:16. Accordingly, a POSITA would have understood that Sampat’s MSM (transform object) automatically controls the flow of Sampat’s source MSP (source object), not that the flow is merely reactive and independent of the MSM. SE1003, ¶¶170-174.

[31.i] providing [61.i] a sink object,

Sampat discloses providing a sink object. SE1003, ¶¶154-165. Under BRI, “sink object” includes “a collection of data and operations that (1) obtains data stream buffers [memory where data can be temporarily stored for transfer] from a transform object and (2) outputs the streams to a video and audio decoder.” SE1014, 91-98; SE1001, 8:52-65.

Sampat describes a sink object in the form of three sink MSPs: the video sink MSP 1614, text sink MSP 1622, and audio sink MSP 1618. SE1004, 9:50-53. These sink MSPs are a collection of data and operations. SE1004 at 9:39-53; SE1003, ¶156.

The server 102 processes data to be (a) multicasted to clients 104 and/or (b) *played locally* on the monitor of the server 102. SE1004, 3:66-4:6, 9:10-26, FIG. 1. The purpose of the sink MSP is for locally playing or recording a data stream. SE1004, 9:39-40 (“A sink MSP is a media service provider that assists in the local playing or recording of a data stream.”). The sink MSPs in the server 102 are used for playing video and audio in substantially the same manner as described with respect to the client 104, for example:

Server software architecture 1512 also preferably has video, audio, and text sink MSPs 1614, 1618, and 1622 to provide ***local monitoring*** capabilities. The processing of sink MSPs is described in further detail

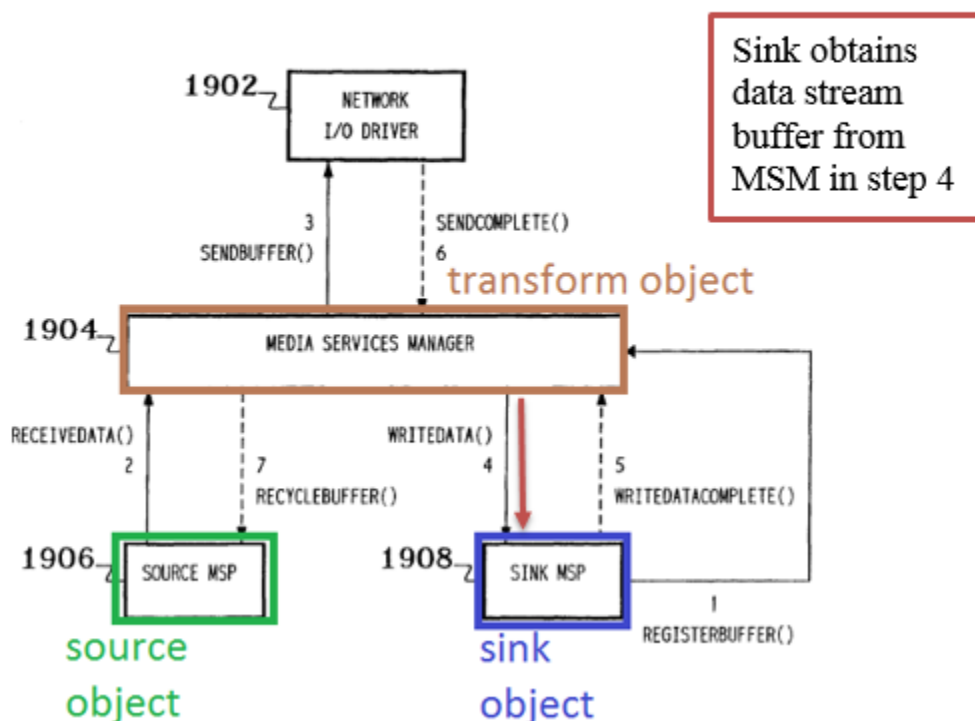
later in this specification *in conjunction with FIG. 18 and the discussion of the client software architecture.*

SE1004, 9:50-57. “Preferably, *each server has all the functionality of a client* to provide monitoring capabilities.” SE1004, 38:34-36 (emphasis added).

Accordingly, the disclosure in Sampat with respect to monitoring (playing video and audio) on the client 104 (*see supra*, Section IV.A) also applies to the server 102. SE1003, ¶¶157-159. The sink MSPs perform each of the two functions defined as being performed by a sink object.

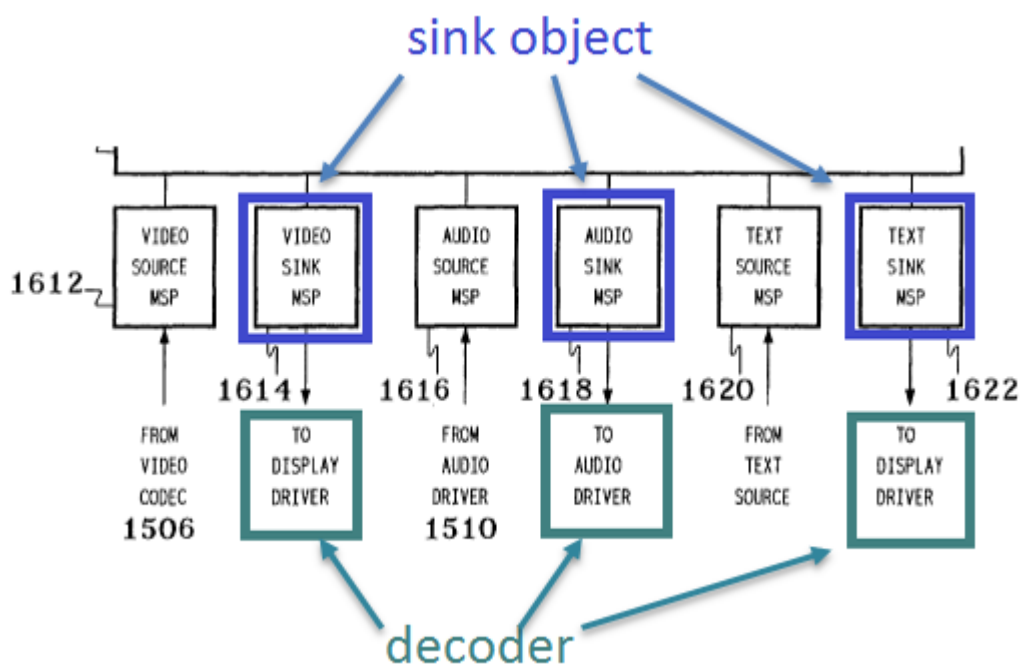
First, Sampat explains that the sink MSPs (sink object) obtain data stream buffers from the MSM (transform object), stating: “MSM 1904 will copy the source buffer data into the next available sink buffer, and write the sink buffer to be played by sink MSP 1908.” SE1004, 17:24-27, *see also* 14:64-15:3. The sink MSP obtaining buffer data from the MSM is shown in FIG. 19, reproduced below.

FIG. 19



SE1004, FIG. 19 (annotated), *see also* 16:60-17:27 (describing “data ... flow from the MSM 1904 to a sink MSP 1908” in that “MSM 1904 will copy the source buffer data into the next available sink buffer, and write the sink buffer to be played by sink MSP 1908”). Dr. Strawn confirms that a POSITA would have understood the sink MSPs as receiving the data stream buffers from the MSM. SE1003, ¶¶160-161.

Second, Sampat describes the sink MSPs outputting the data streams to a video and audio decoder. SE1003, ¶162. For example, Figure 16 of Sampat shows the video sink MSP and audio sink MSP outputting streams to display and audio drivers.



SE1004, FIG. 16 (annotated). Sampat explains that the sink MSPs of the server function similarly to those of the client, for which Sampat explains transmits the video and audio data to respective display and audio drivers. SE1004, 14:64-15:3; SE1003, ¶163.

As explained above at claim element [31.i, 61.i] of Ground 1 (*see, supra*, Section IV.A), a POSITA would have understood that the system of Sampat includes audio and video decoders and that the sink object would output to the audio and video decoders when monitoring on the client 104. SE1004, 4:6-12, 4:56-57, 15:1-3; SE1007, 8, 157-211; SE1021, 2; SE1006, 146, 152-153, 10; SE1001, 4:23-29, 6:63-65. FIG. 7; SE1003, ¶164. Because “each server has all the functionality of a client to provide monitoring capabilities” (SE1004, 38:34-36), a

POSITA would have understood that the system of Sampat includes audio and video decoders and that the sink object would output to the audio and video decoders when monitoring on the server 102 for the same reasons. *Id.*

[31.j, 61.j] wherein said sink object obtains data stream buffers from said transform object and outputs said streams to a video and audio decoder;

As described above at [31.i, 61.i], Sampat discloses this wherein clause, which is identical to language in the construction of sink object as addressed.

SE1003, ¶¶160-164.

[31.k, 61.k] wherein said decoder converts said streams into display signals and sends said signals to a display;

Sampat discloses that the decoder converts the streams into display signals and sends the signals to a display. SE1003, ¶¶180-182. Sampat describes the server 102 having a display (*see, e.g.*, FIG. 1) and describes sending signals to the display for monitoring (SE1004, 9:5-6, 9:50-57). “Preferably, each server has all the functionality of a client to provide monitoring capabilities.” SE1004, 38:34-36. As explained above at claim element [31.k, 61.k] in Ground 1, Sampat teaches the decoder converting streams into display signals and sending those signals to a display monitor. SE1004, 14:6-12, FIG. 17; SE1003, ¶96.

[31.l, 61.l] wherein said sink object is automatically flow controlled by said transform object;

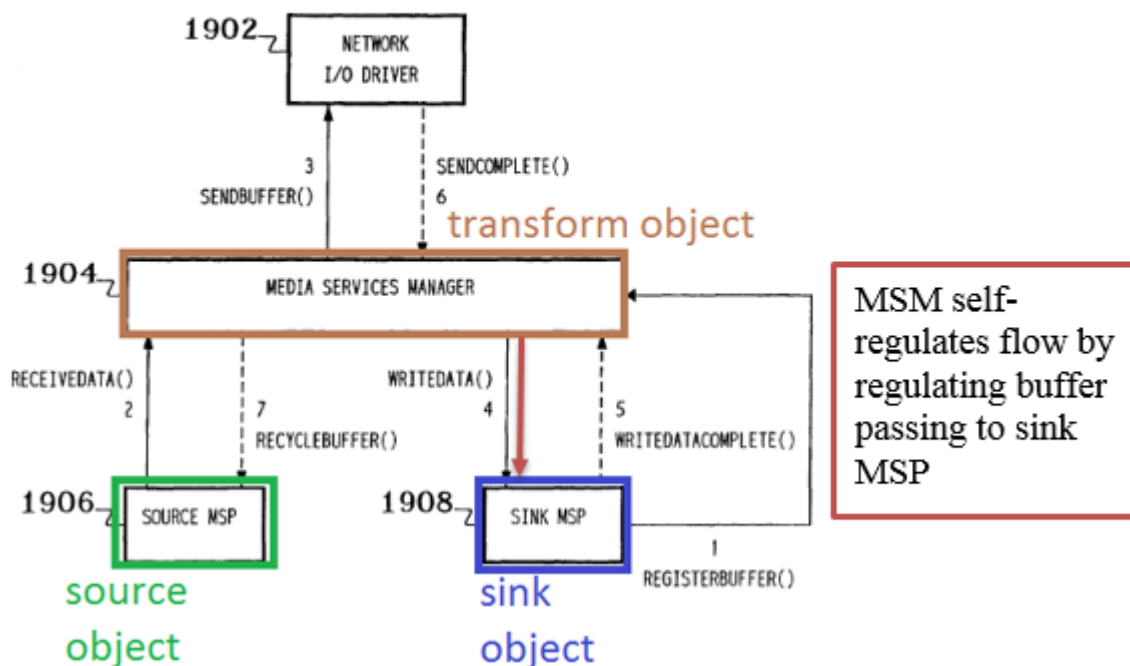
Sampat discloses that the sink object is automatically flow controlled by the transform object. SE1003, ¶¶166-169, 175-179. Under BRI, this claim element

covers “wherein said sink object is *self-regulated* by said transform object.”

SE1014, 101-106; SE1001, 8:52-65. Sampat explains: “Media services manager (MSM) 1608 *manages the flow of data* through server software architecture 1512” including “[f]rom a source MSP to a local sink MSP” and “[f]rom mass storage device 1516 to a local sink MSP.” SE1004, 9:10-26.

The MSM regulates the flow of data from the source object both by starting and stopping flow (SE1004, 9:57-10:29) and also by regulating buffers passing during flow (SE1004, 16:60-17:7). Sampat shows and describes its transform object (MSM) controlling flow in the same way as described in the ’389 patent, with the sink object (sink MSP) playing buffers only when received from the MSM. SE1004 at 16:60-17:39; FIG. 19.

FIG. 19



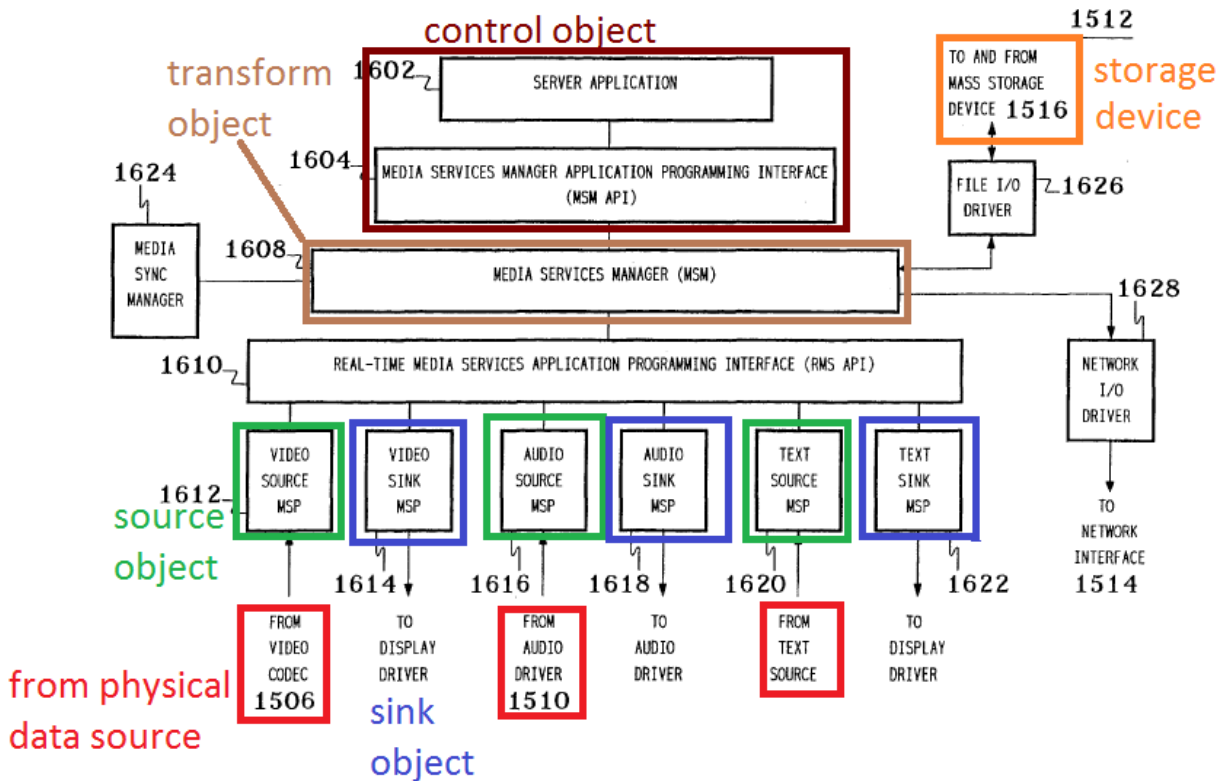
SE1004 at FIG. 19 (annotated), *see also* 17:24-27 (describing the MSM writing the sink buffer data to the sink MSP in step 4 before the sink MSP can play the sink buffer data in step 5); SE1003, ¶177. This description in Sampat is consistent with the description of automatic flow control in the '389 patent, which equates waiting for buffers from the transform object to being “self-regulating” and “automatic flow control.” SE1001 at 8:45-51; *see also* 8:19-9:16. Accordingly, a POSITA would have understood that Sampat teaches the MSM (transform object) automatically controlling the flow of the sink object, not that the flow is merely reactive and independent of the MSM. SE1003, ¶¶178-179.

[31.m] providing [61.m] a control object,

Sampat discloses providing a control object. SE1003, ¶¶183-192. Under BRI, the term “control object” includes “a collection of data and operations that receives commands from a user that control the flow of broadcast data.” SE1014, 92, 98.

Sampat discloses a control object in the form of the server application 1602 and the media services manager (MSM) application programming interface (API) 1604. SE1004, 8:63-9:6 (“server application 1602 is used to select... whether to play any of the selected data streams locally to monitor the multicasting services.”), FIG. 16. Server application 1602 and MSM API 1604 are a collection of data and operations. SE1004, 8:63-9:6.

FIG. 16



SE1004, FIG. 16 (annotated); SE1003, ¶184.

Sampat explains that the combination of server application 1602 and MSM API 1604 controls the flow of broadcast data, stating: “Media services manager (MSM) 1608 *manages the flow* of data through server software architecture 1512 *as specified by server application 1602.*” SE1004, 9:10-26. “Preferably, *each server has all the functionality of a client* to provide monitoring capabilities.” SE1004, 38:34-36 (emphasis added). Accordingly, the disclosure in Sampat with respect to receiving user input on the client 104 (*see supra*, Section IV.A) also applies to the server 102. SE1003, ¶185. The control object (client application

1802 in the client, and server application 1602 in the server) uses the user interface shown in FIGS. 2-14 to control the flow of broadcast data through the system.

SE1004, 14:34-27, 2:17-19, 39:51-54, 7:28-31. Accordingly, a POSITA would have understood that Sampat's server application/MSM API controls the flow of broadcast data through the system by, for example, playing, pausing, recording, and rewinding. SE1003, ¶¶185-187.

[31.n, 61.n] wherein said control object receives commands from a user, said commands control the flow of the broadcast data through the system; and

As described above at [31.m, 61.m], Sampat discloses this wherein clause, which is identical to language in the construction of control object as addressed.

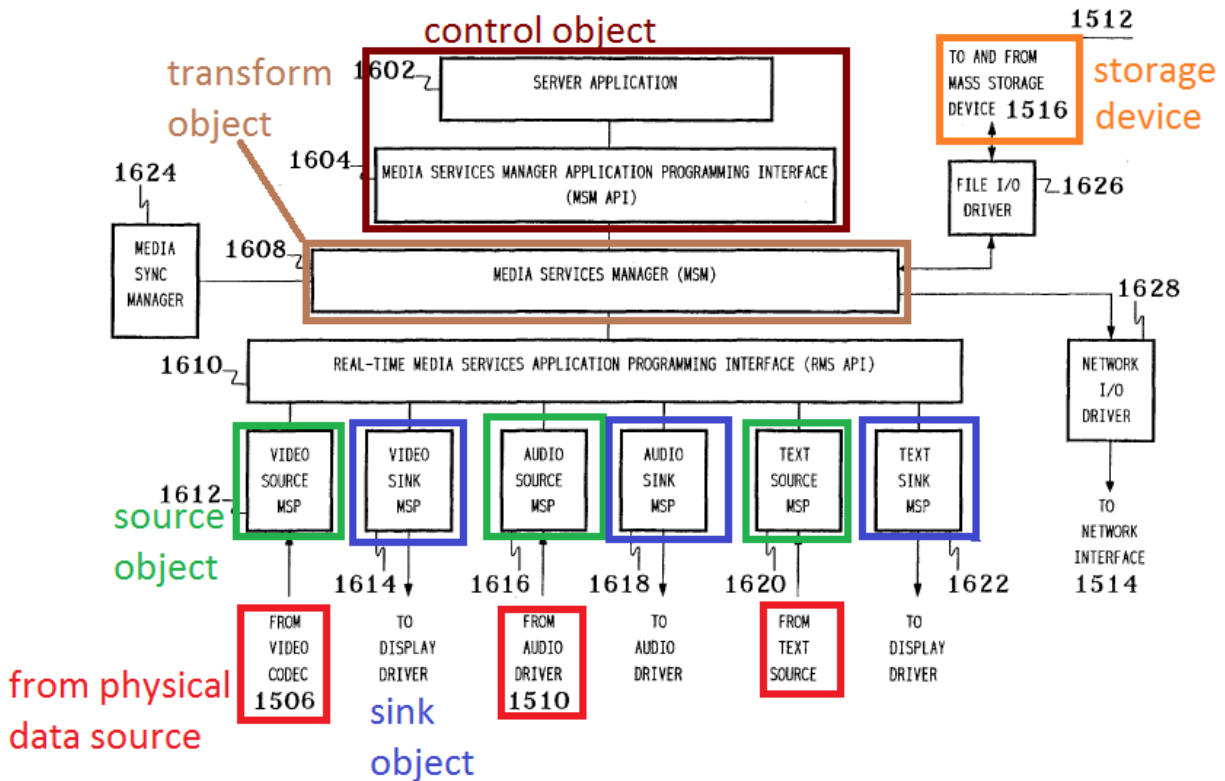
SE1003, ¶¶184-187.

[31.o, 61.o] wherein said control object sends flow command events to said source, transform, and sink objects.

Sampat discloses that the control object sends flow command events to the source, transform, and sink objects. SE1003, ¶¶188-192. Under BRI, this claim element includes "the control object sends information relating to a change of condition in the flow of the broadcast data to the source, transform and sink objects." SE1013, 16-17.

Sampat explains that the control object (server application 1602 and MSM API 1604) sends flow command events to the transform object (MSM) and also to the source and sink objects via the MSM. SE1004, 8:63-9:33, 9:57-10:7.

FIG. 16



SE1004, FIG. 16 (annotated); SE1003, ¶189.

Sampat describes a number of examples of the control object sending information regarding a change of condition, such as starting, pausing, and shutting down. SE1004, 9:57-10:7. In one example of starting a function call via the server application 1602 “Starts (or unpauses) any or all of the MSPs that were initialized.” SE1004, 9:66-67; SE1003, ¶190.

This claim element does not require the flow command events to pass *directly* to each object (under BRI), only that the information is sent to each object. SE1013, 16-17. For example, the ’389 patent describes flow command events

going first to one object and then being relayed to other objects. SE1001, 8:21-24 (“relayed on to the next object and so on down the pipeline.”). Accordingly, a POSITA would have understood that Sampat discloses the control object sending flow command events to the source, transform, and sink objects. SE1003, ¶191.

Therefore, Sampat teaches each and every element of both of claims 31 and 61. SE1003, ¶¶120-192.

D. GROUND 4: Claims 31 and 61 are Obvious under § 103 over Sampat (server-side) in view of VfW, SoundBlaster, and Gerber

As described above in Section IV.C, *supra*, Sampat discloses its server 102 having each and every element of claims 31 and 61. To the extent that Sampat is found not to disclose one or more claim elements, such claim elements would have been obvious in view of the predictable combination of Sampat, VfW, SoundBlaster, and Gerber. SE1003, ¶¶193-217. This is particularly true for claim elements [31.b], [31.i]-[31.k], [61.b], and [61.i]-[61.k].

For example, to the extent that Sampat does not *explicitly* disclose the physical data source temporarily storing the video and audio data (*see* claim elements [31.b] and [61.b]), such functionality was commonly known. SE1003, ¶194. For storing audio data, SoundBlaster discloses storing to memory by describing “Direct Memory Access” (or “DMA”) for use when recording. SE1006, 20, 146, 153, 158. SoundBlaster discloses temporarily storing audio data

when being processed in an audio capture component. SE1006, 98 (“voice is recorded directly to the diskette or hard disk using a double-buffering technique”), 92 (“When you record a sound with the Sound Blaster Pro, it changes analog signals into digital signals.”); SE1003, ¶194.

Multiple reasons would have prompted to a POSITA to implement temporarily storing audio data in an audio capture component (as suggested by SoundBlaster) in the system of Sampat.

First, a POSITA would have been prompted to implement the well-known functionality of temporarily storing audio data in an audio capture component (as suggested by SoundBlaster) in the system of Sampat because temporarily storing data was a traditional function performed when capturing audio data. SE1003, ¶197. A POSITA would have understood that Sampat’s system “captures and converts the analog audio signals into digital audio data streams” and SoundBlaster disclosed a suitable and well-known option for capturing and converting analog audio signals. SE1003, ¶197; SE1004, 7:63-65; SE1006, 20, 92, 98, 146, 153, 158.

Second, a POSITA would have been prompted to implement the functionality of temporarily storing audio data in an audio capture component (as suggested by SoundBlaster) in the system of Sampat because Sampat specifies that “[a]udio capture component 1508 may be any suitable device for capturing and digitizing analog audio signals and *is preferably a Creative Labs SoundBlaster*

Pro.” SE1004, 4:56-57. SoundBlaster captures and digitizes audio signals, thus temporarily storing the signals. SE1006, 1020, 92, 98, 146, 153, 158; SE1003, ¶198.

Third, a POSITA would have been prompted to implement the well-known prior art functionality of temporarily storing audio data in an audio capture component (as suggested by SoundBlaster) in the system of Sampat because doing so would be merely the application of a known technique (e.g., temporarily storing data via an audio capture device like SoundBlaster) to a known system (e.g., Sampat’s system having sound enabled) ready for improvement to yield predictable results. SE1003, ¶199; *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. at 417. Sampat and SoundBlaster disclose the same SoundBlaster device and a POSITA would have recognized that applying the suggestions in SoundBlaster to Sampat’s system would have led to predictable results without significantly altering or hindering the functions performed by the system of Sampat. SE1003, ¶199.

To the extent that Sampat does not *explicitly* disclose that the physical data source temporarily stores the video data (*see* claim elements [31.b] and [61.b]), such functionality was also commonly known in similar prior art capture components. SE1003, ¶200. For example, Gerber describes the “INTEL® SMART VIDEO RECORDER™,” which is the same “Intel® SmartVideo® Recorder (ISVR)” described in Sampat. SE1005, 5:10-16, 5:25-28; SE1004, 8:24-

28. Gerber explains that the Smart Video Recorder include “dynamic random access memory 41 and video random access memory ‘VRAM’) 38.” SE1005, 5:20-28. “In response to a control signal by the CC driver 34, the digitizer 36 *captures* the video data from the video input device 18 and *stores* the captured video data in a first storage element 39a” of the VRAM. SE1005, 5:28-33 (emphasis added), *see also* 5:33-60, FIG. 3.

Multiple reasons would have prompted to a POSITA to implement temporary storing of video data in a video capture component (as suggested by Gerber) in the system of Sampat.

First, a POSITA would have been prompted to implement the well-known prior art functionality of temporarily storing video data in a video capture component (as suggested by Gerber) in the system of Sampat because temporarily storing data was a traditional function performed when capturing video data. SE1003, ¶202. A POSITA would have understood that the video capture component in Sampat’s system “captures and converts the analog video signals into digital video data streams” and Gerber disclosed a suitable and well-known option for capturing and converting analog audio signals. SE1003, ¶202; SE1004, 7:62-63; SE1005, 5:10-16, 5:20-60, FIG. 3.

Second, a POSITA would have been prompted to implement the well-known prior art functionality of temporarily storing video data in a video capture

component (as suggested by Gerber) in the system of Sampat because Sampat specifies that the “[v]ideo capture component 1504 and codec 1506 may be any suitable hardware/software device or devices for capturing and compressing video and are preferably components of an Intel® SmartVideo® Recorder (ISVR).”

SE1004, 8:25-27. The Smart Video Recorder captures and compresses video—temporarily storing the signals. SE1005, 5:10-16, 5:20-60, FIG. 3; SE1003, ¶203. This disclosure in Sampat that the Smart Video Recorder is preferred would have motivated a POSITA to use that very device. SE1003, ¶203.

Third, a POSITA would have been prompted to implement the well-known prior art functionality of temporarily storing video data in a video capture component (as suggested by Gerber) in the system of Sampat because doing so would be merely the application of a known technique (e.g., temporarily storing data via a video capture device like the Intel Smart Video Recorder) to a known system (e.g., Sampat's system having video enabled, preferably by the Intel Smart Video Recorder) ready for improvement to yield predictable results. SE1003, ¶204; *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. at 417. Sampat and Gerber disclose a device for video processing (the same Intel Smart Video Recorder device), and a POSITA would have recognized that applying the suggestions in Gerber to Sampat's system would have led to predictable results without significantly altering or hindering the functions performed by the system of Sampat. SE1003, ¶204.

As explained above in Ground 2 with respect to claim elements [31.i]-[31.k] and [61.i]-[61.k] for the client-side of Sampat, Sampat teaches that the display driver "is preferably Microsoft Video for Windows," and a POSITA would have been prompted to implement the well-known prior art functionality of including a video decoder in the display driver (as suggested by VfW) in the system of Sampat. SE1003, ¶¶108-112, 205-209; SE1004, 13:55-57, 14:6-12; SE1007, 8, 157-211; SE1021, 2. Additionally, Sampat also teaches that "[s]erver 102 and

clients 104 are preferably sound enabled with a SoundBlaster Pro from Creative Labs,” and a POSITA would have been prompted to implement the well-known prior art functionality of outputting to an audio decoder (as suggested by SoundBlaster) in the system of Sampat. SE1003, ¶210; SE1004, 4:56-57, 15:1-15:3; SE1006, 10, 146, 152-153. Because “each server has all the functionality of a client to provide monitoring capabilities” (SE1004, 38:34-36), these teachings also apply to monitoring via the server 102. SE1003, ¶¶164; 210-214. Accordingly, a POSITA would have been motivated to modify Sampat (as already modified by Gerber and SoundBlaster as discussed above) according to the predictable and conventional teachings of VfW and SoundBlaster to provide a sink object that obtains data stream buffers from the transform object and outputs the streams to a video and audio decoder. SE1003, ¶¶193-217.

V. Secondary Considerations

Petitioner is aware that Patent Owner has previously argued that secondary considerations are relevant when considering obviousness of the '389 patent, but such considerations are not impactful to the grounds presented in this petition. SE1003, ¶¶218-224. First, secondary considerations are not relevant to anticipation, and thus are not relevant to Grounds 1 and 3. Second, the alleged secondary considerations have no nexus to claims 31 and 61, but rather, appear to be driven by unclaimed factors. SE1003, ¶219-223; SE1017, 1080, 1083-1084,

1096-1097, 1100, 1118, 1301-1303; *see also* IPR2015-00266, Paper 7 at 22, *citing In re GPAC*, 57 F.3d 1573, 1580 (Fed. Cir. 1995). Third, the evidence shows that the alleged secondary considerations at most relate to the transform object—a feature wholly taught by Sampat—with no allegation of secondary considerations relating to the temporary storage and decoder features being combined in Grounds 2 and 4. SE1003, ¶222; SE1017, 1302-1303. Fourth, the obviousness case is simple, straightforward, and driven by Sampat’s explicit teachings. Thus, TiVo’s secondary considerations evidence would not overcome the strong obviousness grounds, even if it was tied to the features identified as obvious, which it is not. SE1003, ¶224. Indeed, Patent Owner cited Sampat in the reexamination only *after* it was too late for consideration, and therefore, Sampat’s teachings have never been considered by the USPTO in light of TiVo’s secondary consideration evidence or at all. SE1017, 1359-1474, 1481. Accordingly, none of the alleged secondary considerations impact the obviousness grounds based on Sampat.

VI. Redundancy

Petitioner submits that the grounds presented in this Petition are not redundant in that each relies on different structure (Grounds 1-2 vs. Grounds 3-4), different statutes (§102 vs. §103), and/or different references (Ground 2 vs. Ground 4).

VII. CONCLUSION

Claims 31 and 61 of the '389 patent are invalid as set forth in Grounds 1-4.

Petitioner authorizes charge of necessary fees to Deposit Acct. 06-1050.

Respectfully submitted,

Dated: August 2, 2016

/W. Karl Renner/
W. Karl Renner, Reg. No. 41,265
Fish & Richardson P.C.
P.O. Box 1022
Minneapolis, MN 55440-1022
T: 202-626-6447
F: 202-783-2331

(Trial No. IPR2016-01524)

Attorneys for Petitioner

VIII. MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1)

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)

Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc. are the real parties-in-interest.

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

Patent Owner filed a complaint alleging infringement of the '389 patent in a lawsuit against Petitioner in the Eastern District of Texas (*TiVo Inc. v. Samsung Electronics Co., LTD., et al.*, Civil Action No. 2:15-cv-01503). The complaint was filed 9/8/2015.

The '389 patent has also been asserted in at least the following lawsuits: *Cisco Systems, Inc. v. TiVo Inc.* (2:12-cv-00434), *TiVo Inc. v. Cisco Systems, Inc.* (2:12-cv-00311), *Cisco Systems, Inc. v. TiVo Inc.* (3:12-cv-02766), *TiVo Inc. v. Verizon Communications, Inc.* (2:09-cv-00257), *TiVo Inc. v. AT&T Inc.* (2:09-cv-00259), *Dish Network Corporation et al v. TiVo Inc.* (2:09-cv-00171), *Dish Network Corporation et al v. TiVo Inc.* (1:08-cv-00327), and *TiVo Inc. v. EchoStar Communications Corp.* (2:04-cv-00001). The '389 patent was also the subject of two reexamination proceedings, Ser. Nos. 90/007,750 and 90/009,329. Other than Civil Action No. 2:15-cv-01503, Petitioner was not a party to any of these proceedings.

C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Petitioner provides the following designation of counsel.

Lead Counsel	Backup counsel
W. Karl Renner, Reg. No. 41,265 3200 RBC Plaza 60 South Sixth Street Minneapolis, MN 55402 T: 202-783-5070 F: 877-769-7945 Email: IPR39843-0037IP1@fr.com	Jeremy Monaldo, Reg. No. 58,680 Stuart A. Nelson, Reg. No. 63,947 3200 RBC Plaza 60 South Sixth Street Minneapolis, MN 55402 T: 202-783-5070 F: 877-769-7945 Email: PTABInbound@fr.com

D. Service Information

Please address all correspondence and service to the address listed above.

Petitioner consents to electronic service by email at IPR39843-0037IP1@fr.com
(referencing No. IPR39843-0037IP1 and cc'ing PTABInbound@fr.com,
renner@fr.com, monaldo@fr.com, and snelson@fr.com).

Respectfully submitted,

Dated: August 2, 2016

/W. Karl Renner/

W. Karl Renner, Reg. No. 41,265
Fish & Richardson P.C.
P.O. Box 1022
Minneapolis, MN 55440-1022
T: 202-626-6447
F: 202-783-2331

(Trial No. IPR2016-01524)

Attorneys for Petitioner

CERTIFICATION UNDER 37 CFR § 42.24(d)

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for Inter Partes Review totals 13,955, which is less than the 14,000 allowed under 37 CFR § 42.24(a)(i).

Respectfully submitted,

Dated: August 2, 2016

/W. Karl Renner/
W. Karl Renner, Reg. No. 41,265
Fish & Richardson P.C.
P.O. Box 1022
Minneapolis, MN 55440-1022
T: 202-626-6447
F: 202-783-2331

Attorneys for Petitioner

CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§ 42.6(e)(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on August 2, 2016, a complete and entire copy of this Petition for *Inter Partes* Review and all supporting exhibits were provided via Federal Express, to the Patent Owner by serving the correspondence address of record as follows:

HICKMAN PALERMO BECKER BINGHAM LLP
1 ALMADEN BOULEVARD, FLOOR 12
SAN JOSE CA 95113

/Diana Bradley/
Diana Bradley
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
(858) 678-5667