CONTAINS PROTECTIVE ORDER INFORMATION

U.S. Patent No. 8,942,252	Application of Claim Language to Source Code
Claim 1	

[1 Preamble] A method, comprising:	**Source Code Folders:**
	
	`SOFTWARE_STRUCTURE = \test\demo\rules\`
	
	`SOFTWARE_CODE = \beads\`
	
	The Implicit Source Code specifies a distributed system consisting of devices that are nodes of a network ("devices"). These devices execu synchronizing rendering of content provided by a source, where each a rendering time corresponding to the device time when the device re

[1a] a master rendering device rendering a first content stream; and;	Implicit source code specifies a distributed system consisting of a plu These devices include a master rendering device that is set up to rend video content stream. This audio and video content stream rendered rendering device corresponds to a first content stream. Implicit source code implements a distributed system consisting of a devices. An architecture of one such distributed system comprising a devices is defined in files `videomulti.rule`, `videoclient.r` `ipaqvideo.rule`, `pcmaudioserver.rule`, `syncaudio.ru` `timesync.rule`.[1,2] Implicit source code also implements beads o

[1] These files are contained in folder `SOFTWARE_STRUCTURE\`

[2] Another similar distributed system that renders synchronized content streams is described by th folder `\2001.11.01\test\audiosync\`

U.S. Patent No. 8,942,252	Application of Claim Language to Source Code
	on devices that are nodes of a network. Among other services, these perform tasks including decoding and encoding content streams, rece transmitting content streams, synchronizing content streams, and ren streams.[3] Each device is configured using one or more rule files and services implemented by the beads specified within the rules file. Among other devices, the distributed system described in the rules fi master rendering device that renders an audio and video content strea and video content stream rendered by the master rendering device co first content stream. Specifically, the master rendering device receiv audio and video content stream at its local network port 8013 from a device.[5,6] As part of the rendering process, this master rendering dev `avidemux` bead to split the combined audio and video content strea audio and video streams.[7,8] Specifically, functions `AviDemux_EncodeMessageHandler`[9] and `ChunkProcess`[10] `avidemux` separate the combined audio and video content stream. and video streams have been separated, the master rendering device r `bmptorgb` bead to decode the received video content stream to gen

[3] These files are contained in folder `SOFTWARE_CODE\`
[4] Defined at lines 6 to 134 in file `SOFTWARE_STRUCTURE\videomulti.rule`
[5] See line 9 in file `SOFTWARE_STRUCTURE\videomulti.rule`
[6] See lines 7 to 79 in file `\test\demo\source.pl`
[7] See lines 11 to 15 in file `SOFTWARE_STRUCTURE\videomulti.rule`
[8] See file `SOFTWARE_CODE\avidemux\main\avidemux.c`
[9] Implemented at lines 883 to 964 in file `SOFTWARE_CODE\avidemux\main\avidemux.`
[10] Implemented at lines 597 to 849 in file `SOFTWARE_CODE\avidemux\main\avidemux.`

U.S. Patent No. 8,942,252	Application of Claim Language to Source Code
	RGB video frames.[11,12] Specifically, functions `BmpToRgb_Messag` `DecodeFrame`,[14] and `BmpDecoder_FrameDecode`[15] of bead `b` the video content stream to generate the decoded RGB video frames. As discussed earlier, the master rendering device uses the `avidemux` combined audio and video content stream into separate audio and vid the audio and video content stream has been separated, the master ren uses an instance of the `fanout` bead to distribute the audio content s processing pathways.[16] The first processing pathway corresponding t `FanoutIndex 0` outputs the audio content stream using the `spea` audio output device of the master rendering device.[17,18]
[1b] sending, from the master rendering device to a first one of a plurality of slave devices, a plurality of master rendering times	As discussed earlier (*see* Claim 1, Limitation 1a), Implicit source cod distributed system that includes a master rendering device. This mast device receives a combined audio and video content stream sent by a Upon receiving the combined audio and video content stream, the ma device uses the `avidemux` bead to separate the combined audio and

[11] See lines 16 to 20 in file `SOFTWARE_STRUCTURE\videomulti.rule`
[12] See file `SOFTWARE_CODE\bmp2rgb\main\bmp2rgb.c`
[13] Implemented at lines 306 to 335 in file `SOFTWARE_CODE\bmp2rgb\main\bmp2rgb.c`
[14] Implemented at lines 180 to 283 in file `SOFTWARE_CODE\bmp2rgb\main\bmp2rgb.c`
[15] Implemented at lines 366 to 407 in file `SOFTWARE_CODE\bmp2rgb\main\bmpdecoder`
[16] See lines 91 to 97 in file `SOFTWARE_STRUCTURE\videomulti.rule`
[17] See lines 104 to 107 in file `SOFTWARE_STRUCTURE\videomulti.rule`
[18] See file `SOFTWARE_CODE\speaker\main\speaker.c`

U.S. Patent No. 8,942,252	Application of Claim Language to Source Code
indicative of statuses of the rendering the first content stream at the master rendering device at different times;	separate audio and video streams. The `avidemux` bead instantiates rendering clock `IAudioClock` associated with the audio content st master rendering clock `IAudioClock` generates a plurality of mast that are indicative of the of statuses of the rendering the combined au content stream by the master rendering device at different times. Furt source code specifies a distributed system that also includes plurality rendering devices. One of these slave rendering devices renders PCM stream in synchronization with the master rendering device.[20] Anoth rendering devices renders the video content stream in synchronizatio rendering device.[21] The master rendering device sends a plurality of PCM frames and the corresponding master rendering times indicative the rendering of the first content stream by the master rendering devi rendering device rendering audio content stream at port 9002.[22,23] Si master rendering device sends a plurality of encoded video RGB vide corresponding master rendering times indicative of the statuses of the first content stream by the master rendering device to the slave rende rendering video content stream at port 8002.[24,25] As discussed earlier (*see* Claim 1, Limitation 1a), upon receiving the

[19] See lines 447 to 451 in file `SOFTWARE_CODE\avidemux\main\avidemux.c`
[20] Defined at lines 3 to 25 in file `SOFTWARE_STRUCTURE\syncaudio.rule`
[21] Defined at lines 7 to 35 in file `SOFTWARE_STRUCTURE\videoclient.rule`
[22] See lines 123 to 127 in file `SOFTWARE_STRUCTURE\videomulti.rule`
[23] See line 4 in file `SOFTWARE_STRUCTURE\syncaudio.rule`
[24] See lines 123 to 127 in file `SOFTWARE_STRUCTURE\videomulti.rule`
[25] See line 4 in file `SOFTWARE_STRUCTURE\syncaudio.rule`

U.S. Patent No. 8,942,252	Application of Claim Language to Source Code
	and video content stream, the master rendering device uses the `avid` separate the combined audio and video stream into separate audio an While processing the combined audio and video content streams, the calls function `AudioPrepare`.[26] Function `AudioPrepare` initia rendering clock `IAudioClock` associated with the audio content st master rendering clock `IAudioClock` generates a plurality of mast that are indicative of the of statuses of the rendering the combined au content stream by the master rendering device at different times. As discussed earlier (*see* Claim 1, Limitation 1a), the master renderin processes the video content stream using the `bmptorgb` bead to ren video frames. Once the video frame has been decoded into an RGB master rendering device uses a `fanout` bead to distribute the decod two processing pathways or fanouts.[28,29] Specifically, function `FanOut_MessageHandler`[30] of bead `fanout` distributes decod frames to the two processing pathways. Within one of these processi the master rendering device uses the `clocksync` bead to encode the

[26] Implemented at lines 389 to 477 in file `SOFTWARE_CODE\avidemux\main\avidemux.`
[27] See lines 447 to 451 in file `SOFTWARE_CODE\avidemux\main\avidemux.c`
[28] See lines 21 to 25 in file `SOFTWARE_STRUCTURE\videomulti.rule`
[29] See file `SOFTWARE_CODE\fanout\main\fanout.c`
[30] Implemented at lines 180 to 199 in file `SOFTWARE_CODE\fanout\main\fanout.c`
[31] See lines 30 to 49 in file `SOFTWARE_STRUCTURE\videomulti.rule`

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.