US Patent & Trademark Office
Patent Public Search | Text View

United States Patent	12389066
Kind Code	B2
Date of Patent	August 12, 2025
Inventor(s)	Deshpande; Sachin G.

Receiving device, signaling device, and method for receiving recovery file information

Abstract

A system for retrieving watermark associated recovery data information.

Inventors: **Deshpande; Sachin G. (Vancouver, WA)**

Applicant: **Sharp Kabushiki Kaisha** (Sakai, JP)

Family ID: **58796813**

Assignee: **SHARP KABUSHIKI KAISHA (Sakai, JP)**

Appl. No.: **18/524689**

Filed: **November 30, 2023**

Prior Publication Data

Document Identifier	Publication Date
US 20240121470 A1	Apr. 11, 2024

Related U.S. Application Data

continuation parent-doc US 17694791 20220315 US 11924504 child-doc US 18524689
continuation parent-doc US 16908780 20200623 US 11317141 20220426 child-doc US 17694791
continuation parent-doc US 15780667 US 10735802 20200804 WO PCT/JP2016/085448 20161129 child-doc US 16908780
us-provisional-application US 62373696 20160811
us-provisional-application US 62310636 20160318
us-provisional-application US 62302151 20160301
us-provisional-application US 62263520 20151204

Publication Classification

Int. Cl.: **H04N7/173** (20110101); **H04N21/235** (20110101); **H04N21/435** (20110101); **H04N21/4722** (20110101); **H04N21/81** (20110101); **H04N21/8358** (20110101); **H04N21/858** (20110101)

U.S. Cl.:

CPC **H04N21/435** (20130101); **H04N21/235** (20130101); **H04N21/4722** (20130101); **H04N21/812** (20130101); **H04N21/8358** (20130101); **H04N21/8586** (20130101);

Field of Classification Search

CPC: H04N (21/435); H04N (21/235); H04N (21/4722); H04N (21/812); H04N (21/8358); H04N (21/8586)

References Cited

U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
10147433	12/2017	Bradley	N/A	G10L 19/018
2015/0324947	12/2014	Winograd	382/100	G06F 16/9566
2015/0358507	12/2014	Eyer	348/515	H04N 21/8586
2018/0254940	12/2017	Kwak	N/A	H04L 67/51
2019/0007753	12/2018	Yang	N/A	H04N 21/2362
2021/0119853	12/2020	Kwak	N/A	H04L 67/51

OTHER PUBLICATIONS

Deshpande, “Method of Receiving a Recovery File Format”, U.S. Appl. No. 17/694,791, filed Mar. 15, 2022. cited by applicant

---

Primary Examiner: Shang; Annan Q

Attorney, Agent or Firm: Keating & Bennett, LLP

Background/Summary

REFERENCE TO THE RELATED APPLICATION (1) This Non-provisional application claims priority under 35 U.S.C. § 119 on Patent Applications No. 62/263,520 filed on Dec. 4, 2015, No. 62/302,151 filed on Mar. 1, 2016, No. 62/310,636 filed on Mar. 18, 2016, and No. 62/373,696 filed on Aug. 11, 2016, the entire contents of which are hereby incorporated by reference.

TECHNICAL FIELD
(1) The present invention relates generally to a system with audio-visual content watermarking.
BACKGROUND ART
(2) In many digital broadcasting systems, a broadcasting station transmits audio-visual content and one or more enhanced service data. The enhanced service data may be provided with the audio-visual (AV) content to provide information and services or may be provided separately from the AV

content to provide information and services.
(3) In many broadcasting environments, the AV content and the one or more enhanced service data is not received directly by an AV presentation device from the broadcasting station. Rather the AV presentation device, such as a television, is typically connected to a broadcast receiving device that receives the AV content and the one or more enhanced service data in a compressed form and provides uncompressed AV content to the AV presentation device.
(4) In some broadcasting environments, the broadcast receiving device receives AV content from a server (sometimes referred to as a Multichannel Video Programming Distributor (MVPD). The MVPD receives an AV broadcast signal from the broadcasting station, extracts content from the received AV broadcast signal, converts the extracted content into AV signals having a suitable format for transmission, and provides the converted AV signals to the broadcast receiving device. During the conversion process, the MVPD often removes the enhanced service data provided from the broadcasting station or may incorporate a different enhanced service data that is provided to the broadcast receiving device. In this manner, the broadcasting station may provide the AV content with enhanced service data, but the enhanced service data, if any, that is ultimately provided to the AV presentation device and/or the broadcast receiving device may not be the same as that provided by the broadcasting station.

SUMMARY OF INVENTION

Technical Problem

(5) Since the broadcast receiving device extracts AV content from the signal received from the MVPD and provides only uncompressed AV data to the AV presentation device, only enhanced service data provided to the broadcast receiving device is available. Furthermore, the same enhanced service data provided by the broadcasting station may not be provided to the broadcast receiving device and/or AV presentation device.

Solution to Problem

(6) According to one embodiment of the present invention, there is provided a method for receiving a recovery file format file from a provider comprising the steps of (a) receiving a recovery data table including a RecoveryDataTable element; (b) receiving a contentID field of said RecoveryDataTable element describing a type of content identifier provided in a message having a cardinality of 0 . . . N; (c) receiving a svcInetUrl field of said RecoveryDataTable element describing service information; (d) receiving a URLValue field of said svcInetUrl field describing URL to access Internet signaling files for said service information; (e) decoding elements of said file based upon said recovery data table.
(7) According to one embodiment of the present invention, there is provided a receiver receiving a recovery file format file from a provider comprising the steps of (a) said receiver receiving a recovery data table including a RecoveryDataTable element; (b) said receiver receiving a contentID field of said RecoveryDataTable element describing a type of content identifier provided in a message having a cardinality of 0 . . . N; (c) said receiver receiving a svcInetUrl field of said RecoveryDataTable element describing service information; (d) receiving a URLValue field of said svcInetUrl field describing URL to access Internet signaling files for said service information; (e) said receiver decoding elements of said file based upon said recovery data table.

Advantageous Effects of Invention

(8) The foregoing and other objectives, features, and advantages of the invention will be more readily understood upon consideration of the following detailed description of the invention, taken in conjunction with the accompanying drawings.

Description

BRIEF DESCRIPTION OF DRAWINGS

(1) FIG. **1** illustrates a system with enhanced service information.
(2) FIG. **2** illustrates another system with enhanced information.
(3) FIG. **3** illustrates a data flow for a system with enhanced information.
(4) FIG. **4** illustrates another system with enhanced information.
(5) FIG. **5** illustrates a watermark payload.
(6) FIG. **6** illustrates another watermark payload.
(7) FIG. **7** illustrates relationships between watermark payloads.
(8) FIG. **8** illustrates relationships between watermark payloads.
(9) FIG. **9** illustrates relationships between watermark payloads.
(10) FIG. **10** illustrates another system with enhanced information.
(11) FIG. **11** illustrates obtaining synchronization and maintaining synchronization.
(12) FIG. **12** illustrates another watermark payload.
(13) FIG. **13** illustrates a standards development organization (SDO) private data.
(14) FIG. **14** illustrates metadata encapsulated within SDO private data as a SDO payload using one or more cmdIDs.
(15) FIG. **15** illustrates an exemplary JavaScript Object Notation schema.
(16) FIG. **16**A illustrates logical structure of a JavaScript Object Notation schema.
(17) FIG. **16**B illustrates the left half part of FIG. **16**A.
(18) FIG. **16**C illustrates the lower half part of FIG. **16**A.
(19) FIG. **17** illustrates an exemplary watermark associated information retrieval JavaScript Object Notation schema.
(20) FIG. **18** illustrates an exemplary recovery file format JavaScript Object Notation schema.
(21) FIG. **19** illustrates an exemplary watermark associated information retrieval JavaScript Object Notation schema.
(22) FIG. **20** illustrates an exemplary recovery file format JavaScript Object Notation schema.
(23) FIG. **21** illustrates an exemplary recovery file format JavaScript Object Notation schema.
(24) FIG. **22** illustrates an exemplary recovery file format logical structure.
(25) FIG. **23** illustrates an exemplary component description logical structure.
(26) FIG. **24**A illustrate an exemplary component anchor logical structure.
(27) FIG. **24**B illustrate an exemplary component anchor logical structure.
(28) FIG. **24**C illustrate an exemplary component anchor logical structure.
(29) FIG. **25**A illustrates an exemplary recovery file format JavaScript Object Notation schema.
(30) FIG. **25**B illustrates an exemplary recovery file format JavaScript Object Notation schema.
(31) FIG. **25**C illustrates an exemplary recovery file format JavaScript Object Notation schema.
(32) FIG. **25**D illustrates an exemplary recovery file format JavaScript Object Notation schema.
(33) FIG. **26**A illustrates an exemplary recovery file format JavaScript Object Notation schema.
(34) FIG. **26**B illustrates an exemplary recovery file format JavaScript Object Notation schema.
(35) FIG. **26**C illustrates an exemplary recovery file format JavaScript Object Notation schema.
(36) FIG. **26**D illustrates an exemplary recovery file format JavaScript Object Notation schema.
(37) FIG. **27**A illustrates an exemplary recovery file format JavaScript Object Notation schema.
(38) FIG. **27**B illustrates an exemplary recovery file format JavaScript Object Notation schema.
(39) FIG. **27**C illustrates an exemplary recovery file format JavaScript Object Notation schema.
(40) FIG. **27**D illustrates an exemplary recovery file format JavaScript Object Notation schema.
(41) FIG. **28** illustrates an exemplary recovery file format logical structure.
(42) FIG. **29** illustrates an exemplary component description logical structure.
(43) FIG. **30** illustrates an exemplary component anchor logical structure.
(44) FIG. **31** illustrates exemplary slsProtocol values.
(45) FIG. **32** illustrates exemplary urlType values.

(46) FIG. **33**A illustrates an exemplary recovery file format JavaScript Object Notation schema.
(47) FIG. **33**B illustrates an exemplary recovery file format JavaScript Object Notation schema.
(48) FIG. **33**C illustrates an exemplary recovery file format JavaScript Object Notation schema.
(49) FIG. **33**D illustrates an exemplary recovery file format JavaScript Object Notation schema.
(50) FIG. **33**E illustrates an exemplary recovery file format JavaScript Object Notation schema.
(51) FIG. **34**A illustrates an exemplary recovery file format JavaScript Object Notation schema.
(52) FIG. **34**B illustrates an exemplary recovery file format JavaScript Object Notation schema.
(53) FIG. **34**C illustrates an exemplary recovery file format JavaScript Object Notation schema.
(54) FIG. **34**D illustrates an exemplary recovery file format JavaScript Object Notation schema.
(55) FIG. **35**A illustrates an exemplary recovery file format JavaScript Object Notation schema.
(56) FIG. **35**B illustrates an exemplary recovery file format JavaScript Object Notation schema.
(57) FIG. **35**C illustrates an exemplary recovery file format JavaScript Object Notation schema.
(58) FIG. **35**D illustrates an exemplary recovery file format JavaScript Object Notation schema.
(59) FIG. **35**E illustrates an exemplary recovery file format JavaScript Object Notation schema.
(60) FIG. **36**A illustrates an exemplary recovery file format logical structure.
(61) FIG. **36**B illustrates an exemplary recovery file format logical structure.
(62) FIG. **37**A illustrates an exemplary recovery file format JavaScript Object Notation schema.
(63) FIG. **37**B illustrates an exemplary recovery file format JavaScript Object Notation schema.
(64) FIG. **37**C illustrates an exemplary recovery file format JavaScript Object Notation schema.
(65) FIG. **37**D illustrates an exemplary recovery file format JavaScript Object Notation schema.
(66) FIG. **37**E illustrates an exemplary recovery file format JavaScript Object Notation schema.

DESCRIPTION OF EMBODIMENTS

(67) Referring to FIG. **1**, the system may include a content source **100**, a content recognizing service providing server **120**, a MVPD **130**, an enhanced service information providing server **140**, a broadcast receiving device **160**, a network **170**, and an AV presentation device **180**.

(68) The content source **100** may correspond to a broadcasting station that broadcasts a broadcast signal including one or more streams of AV content (e.g., audio and/or video). The broadcasting station may use Advanced Television systems Committee (ATSC) emission specifications. The broadcast signal may further include enhanced services data and/or signaling information. The enhanced services data preferably relates to one or more of the AV broadcast streams. The enhanced data services may have any suitable format, such as for example, service information, metadata, additional data, compiled execution files, web applications, hypertext markup language (HTML) documents, extensible markup language (XML) documents, cascading style sheet (CSS) documents, audio files, video files, ATSC , future versions contents, and addresses such as a uniform resource locator (URL).

(69) The content recognizing service providing server **120** provides a content recognizing service that allows the AV presentation device **180** to recognize content on the basis of AV content from the content source **100**. The content recognizing service providing server **120** may optionally modify the AV broadcast content, such as by including a watermark.

(70) The content recognizing service providing server **120** may include a watermark inserter. The watermark inserter may insert watermarks which are designed to carry enhanced services data and/or signaling information, while being imperceptible or at least minimally intrusive to viewers. In other cases a readily observable watermark may be inserted (e.g., readily observable may be readily visible in the image and/or readily observable may be readily audible in the audio). For example, the readily observable watermark may be a logo, such as a logo of a content provider at the upper-left or upper-right of each frame.

(71) The content recognizing service providing server **120** may include a watermark inserter that modifies the AV content to include a non-readily observable watermark (e.g., non-readily observable may be readily non-visible in the image and/or non-readily observable may be non-readily audible in the audio). For example, the non-readily observable watermark may include

security information, tracking information, data, or otherwise. Another example includes the channel, content, timing, triggers, and/or URL information.

(72) The MVPD **130** receives broadcast signals from one or more broadcasting stations and typically provides multiplexed broadcast signals to the broadcast receiving device **160**. The MVPD **130** may perform demodulation and channel decoding on the received broadcast signals to extract the AV content and enhanced service data. The MVPD **130** may also perform channel encoding on the extracted AV content and enhanced service data to generate a multiplexed signal for further distribution. The MVPD **130** may exclude the extracted enhanced service data and/or may include a different enhanced service data.

(73) The broadcast receiving device **160** may tune to a channel selected by a user and receive an AV signal of the tuned channel. The broadcast receiving device **160** typically performs demodulation and channel decoding on the received signal to extract desired AV content. The broadcast receiving device **160** decodes the extracted AV content using any suitable technique, such as for example, a H.264, a Motion Picture Experts Group (MPEG) Advanced Video Coding (AVC), an H.265, a High Efficiency Video Coding (HEVC), a Dolby Digital (AC3), and/or an Advanced Audio Coding (AAC) system. The broadcast receiving device **160** typically provides uncompressed AV content to the AV presentation device **180**.

(74) The enhanced service information providing server **140** provides enhanced service information to AV content in response to a request from the AV presentation device **180**.

(75) The AV presentation device **180** may include a display, such as for example, a television, a notebook computer, a mobile phone, and a smart phone. The AV presentation device **180** may receive uncompressed (or compressed) AV or video or audio content from the broadcast receiving device **160**, a broadcast signal including encoded AV or video or audio content from the content source **100**, and/or encoded or decoded AV or video or audio content from the MVPD **130**. In some cases the uncompressed video and audio may be received via an High-Definition Multimedia Interface (HDMI) cable. The AV presentation device **180** may receive from the content recognizing service providing server **120** through the network **170**, an address of an enhanced service relating to the AV content from the enhanced service information providing server **140**.

(76) It is to be understood that the content source **100**, the content recognizing service providing server **120**, the MVPD **130**, and the enhanced service information providing server **140** may be combined, or omitted, as desired. It is to be understood that these are logical roles. In some case some of these entities may be separate physical devices. In other cases some of these logical entities may be embodied in same physical device. For example, the broadcast receiving device **160** and AV presentation device **180** may be combined, if desired.

(77) Referring to FIG. **2**, a modified system may include a watermark inserter **190**. The watermark inserter **190** may modify the AV (e.g., the audio and/or video) content to include additional information in the AV content. The MVPD **130** may receive and distribute a broadcast signal including the modified AV content with the watermark.

(78) The watermark inserter **190** preferably modifies the signal in a manner that includes additional information which is non-readily observable (e.g., visually and/or audibly) in the form of digital information. In non-readily observable watermarking, the inserted information may be readily identifiable in the audio and/or video. In non-readily observable watermarking, although information is included in the AV content (e.g., the audio and/or video), a user is not readily aware of the information.

(79) One use for the watermarking is copyright protection for inhibiting illegal copying of digital media. Another use for the watermarking is source tracking of digital media. A further use for the watermarking is descriptive information for the digital media. Yet another use for the watermarking is providing location information for where additional content may be received associated with the digital media. Yet another use is to identify content and content source that is being viewed and the current time point in the content, and then allowing the device to access the desired additional

functionality via an Internet connection. The watermark information is included within the AV content itself, as distinguished from, meta-data that is delivered along with the AV content. By way of example, the watermark information may be included by using a spread spectrum technique, a quantization technique, and/or an amplitude modulation technique.

(80) Referring to FIG. **3**, an exemplary data flow is illustrated. The content source **100** transmits a broadcast signal including at least one AV content and an enhanced service data **201** to the watermark inserter **190**.

(81) The watermark inserter **190** receives the broadcast signal that the content source **100** provides and includes a readily observable and/or a non-readily observable watermark in the AV content. The modified AV content with the watermark is provided together with enhanced service data **203** to the MVPD **130**.

(82) The content information associated with the watermark may include, for example, identification information of a content provider that provides AV content, AV content identification (ContentID) information, time information of a content section used in content information acquisition, names of channels through which AV content is broadcasted, logos of channels through which AV content is broadcasted, descriptions of channels through which the AV content is broadcasted, a usage information reporting period, the minimum usage time for usage information acquisition, statistics for sporting events, display of useful information, widgets, applications, executables, and/or available enhanced service information relating to AV content.

(83) The acquisition path of available enhanced service data may be represented in any manner, such an Internet Protocol (IP) based path or an ATSC-Mobile Handheld path.

(84) The MVPD **130** receives broadcast signals including watermarked AV content and enhanced data service and may generate a multiplexed signal to provide it **205** to the broadcast receiving device **160**. At this point, the multiplexed signal may exclude the received enhanced service data and/or may include a different enhanced service data.

(85) The broadcast receiving device **160** may tune to a channel that a user selects and receives signals of the tuned channel, demodulates the received signals, performs channel decoding and audio-video decoding on the demodulated signals to generate an uncompressed audio-video content, and then, provide **206** the uncompressed AV content to the AV presentation device **180**. The content source **100** may also broadcast **207** the AV content through a channel to the AV presentation device **180**. The MVPD **130** may directly transmit **208** a broadcast signal including AV content to the AV presentation device **180** without going through the broadcast receiving device **160**. In yet another case some of the AV information may be sent to the AV presentation device **180** over a broadband connection. In some cases this may be managed broadband connection. In another case it may be unmanaged broadband connection.

(86) The AV presentation device **180** may receive uncompressed (or compressed) AV content from the broadcast receiving device **160**. Additionally, the AV presentation device **180** may receive a broadcast signal through a channel from the content source **100**, and then, may demodulate and decode the received broadcast signal to obtain AV content. Additionally, the AV presentation device **180** may receive a broadcast signal from the MVPD **130**, and then, may demodulate and decode the received broadcast signal to obtain AV content. The AV presentation device **180** (or broadcast receiving device **160**) extracts watermark information from one or more video frames or a selection of audio samples of the received AV content. The AV presentation device **180** may use the information obtained from the watermark(s) to make a request **209** to the enhanced service information providing server **140** (or any other device) for additional information. The enhanced service information providing server **140** may provide, in response thereto a reply **211**.

(87) Referring to FIG. **4**, a further example includes the content source **100** that provides AV content together with enhanced service data (if desired) to the watermark inserter **190**. In addition, the content source **100** may provide a code **300** to the watermark inserter **190** together with the AV content. The code **300** may be any suitable code to identify which, among a plurality of AV

streams, should be modified with the watermark. For example code=1 may identify the first AV stream, code=2 may identify the second AV stream, code=3 may identify the third AV stream, code=4 may identify the fourth AV stream, etc. The code may include temporal location information within the AV content. The code may include other metadata, if desired.

(88) The watermarked AV content and associated data, signaling is provided by the watermark inserter **190** to the MVPD, which in turn may provide the watermarked compressed AV content to the broadcast receiving device **160** (e.g., a set top box). The broadcast receiving device **160** may provide watermarked AV content (e.g., typically uncompressed) to the AV presentation device **180**. The AV presentation device **180** may include a watermark capable receiver **310** together with a watermark client **320**. The watermark capable receiver **310** is suitable to detect the existence of the watermark within the AV content, and to extract the watermark data from within the AV content. The watermark client **320** is suitable to use the data extracted from the watermark to request additional data based thereon, and subsequently use this additional data in a suitable manner.

(89) The AV presentation device **180** may use the code **300** from the extracted watermark to make a request to a metadata server **350**. A code database **370** receives the data from the content source **100** that includes the code **300** and metadata **360**. The code **300** and metadata **360** is stored in the code database **370** for subsequent use. In this manner, the code **300** that is provided to the watermark inserter **190** which is encoded within the AV content is also stored in the code database **370** together with its metadata **360**. In the event that the MVPD **130**, or otherwise, removes the associated metadata or otherwise changes the associated metadata, it is recoverable by the AV presentation device **180** from the metadata server **350** which uses the provided code **351** to query the code database **370** and provide an associated response with the metadata **353** to the AV presentation device **180**. The reply metadata provided by the metadata server **350** is used by the AV presentation device **180** to form a request **355** that is provided to the content and signaling server **380**. The content and signaling server **380**, in response to the request, provides selected content and signaling **357** to the AV presentation device **180**. In general, the content and signaling server **380** may be different from the metadata server **350**.

(90) However, making a first request to the metadata server to obtain a response to the code provided, then subsequently using the metadata to provide a request to the content and signaling server **380** is burdensome, and prone to failure, due to the two different servers and/or requests that are utilized. Additionally it may increase the latency.

(91) By way of example, the metadata may consist of one or more of the following syntax elements: (1) location of content and signaling server (e.g., where is the server, such as its network address. Examples of network addresses are domain names, IP v4 addresses, etc.); (2) protocol to be used for communication with the content and signaling server; for example, the Hypertext Transfer Protocol Secure (HTTPS) or the Hypertext Transfer Protocol (HTTP); (3) time code identifying a temporal location in the AV content (e.g., where the metadata should be associated with in the AV content); (4) time sensitive event trigger (e.g., an advertisement or an event for a particular location in the AV content); (5) channel identification (e.g., channel specific information; local channel content); (6) duration over which the content and signaling server requests are randomly carried out by client (e.g., for load balancing). For brevity, this syntax element may also be referred to as duration for content server requests; (7) etc.

(92) The watermark(s) embedded in the audio-video content typically have a capacity to carry only a few bits of payload information when the watermarked audio-video broadcast has non-readily observable information. For relatively small payload sizes, the time code (element 3 above) and/or the location of the content and signaling server (element 1 above) tends to take on a significant percentage of the available payload leaving limited additional payload for the remaining data, which tends to be problematic.

(93) To include sufficient metadata within the watermark, so that both the time code and the location information may be provided together with additional information, it may be desirable to

partition the metadata across multiple watermark payloads. Each of the watermark payloads is likewise preferably included within different portions of the AV content. The data extracted from the multiple watermark payloads are combined together to form a set of desirable information to be used to make a request. In the description below the term payload may be used to indicate watermark payload. Each of the syntax elements may be included within a single payload, spanned across multiple payloads, and/or fragmented across multiple payloads. Each payload may be assigned a payload type for purposes of identification. Further, an association may be established between multiple payloads belonging to the same or approximately the same timeline location. Also, the association may be uni-directional or bi-directional, as desired.

(94) The desired time code data may be obtained from payload(s) that span across several temporal locations of the AV content. Therefore some systems may establish rules to associate the determined time code with a particular temporal location of the AV content. In an example, the chosen temporal location may correspond to the temporal location at the end of a pre-determined watermark payload.

(95) For example, the payload size may be 50 bits while the desirable metadata may be 70 bits, thus exceeding the payload size of a single watermark. An example of the desirable metadata may be as follows:

(96) TABLE-US-00001 location of content and server (I) 32 bits (IP address) application layer protocol (A) 1 bit (HTTP or HTTPS) time code (T) 25 bits (for 1 year of uniqueness with a granularity of 1 second) time sensitive trigger (D) 1 bit (A value of 1 indicates the AV presentation device should query for interactive content. A value of 0 indicates the AV presentation device should not query for interactive content (e.g. as in time base trigger)). channel identification (L) 9 bits duration for content server 2 bits requests (R)

(97) Another example of the desirable metadata may be as follows:

(98) TABLE-US-00002 location of content and server (I) 32 bits (IP address) application layer protocol (A) 2 bit (00 = HTTP, 01 = HTTPS, 10 = reserved, 11 = reserved) time code (T) 25 bits (for 1 year of uniqueness with a granularity of 1 second) time sensitive trigger (D) 1 bit channel identification (L) 9 bits duration for content server 2 bits requests (R)

(99) One manner of partitioning the metadata is to include the content and signal server communication information (CSSCI) in one payload and timeline information in another payload. The CSSCI payload may include, for example, where information (e.g., location of content and signaling server), association information (e.g., an identifier to associate the CSSCI payload with one or more other payloads), and how information (e.g., application layer protocol, duration for content server requests). The timeline information may include, for example, association information (e.g., an identifier to associate the timeline with one or more other payloads), when information (e.g., time code information), and which information (e.g., channel identification).

(100) Referring to FIG. **5**, an exemplary CSSCI payload is illustrated.

(101) Referring to FIG. **6**, an exemplary time location payload is illustrated. The term time location may be alternatively used in place of the term temporal location.

(102) The payload type may be identified by the first bit, “Y”. When Y is set to 0 the payload corresponds to CSSCI payload and the 14 bit payload identifier (P) is used to label the CSSCI. When Y is set to 1 the payload corresponds to the temporal location payload and the 14 bit payload identifier (P) signals the corresponding CSSCI. As a result, different payload types with same payload identifier (P) value are associated with each other. The identifier R indicates a time duration over which to spread the content and signaling server requests. In an example, Y may correspond to a 2-bit field where the value 00 indicates a CSSCI payload, the value 01 indicates a temporal location payload and the values 10, 11 are reserved for future use.

(103) Referring to FIG. **7**, an exemplary time line is illustrated. A first CSSCI type payload (e.g., CSSCI-0) has a first set of association information P while a second CSSCI type payload (e.g., CSSCI-1) has a second different set of association information P. Having two different association

information P for CSSCI-0 and CSSCI-1 distinguish between and identify the two CSSCI payloads. A first time location payload (e.g., Timeline-0) has the first set of association information P that matches the association information P for CSSCI-0, a second time location payload (e.g., Timeline-1) has the same first set of association information P that matches the association information P for CSSCI-0, a third time location payload (e.g., Timeline-2) has the same second set of association information P that matches the association information P for CSSCI-1. In this manner, CSSCI-0, Timeline-0; CSSCI-0, Timeline-1; and CSSCI-1, Timeline-2 are associated together as pairs having spanned watermarked information. This permits the same CSSCI type payload to be used for multiple different time location payloads.

(104) As illustrated, each temporal location payload is associated with a previously received CSSCI type payload, and thus unidirectional in its association. In the event that a previous CSSCI type payload matching a temporal location payload is not available, then the system may be able to determine that a packet has been lost or otherwise the watermarking was not effective. The loss of watermarking data occurs with some frequency because the audio-video content tends to be modified by audio-video transcoding, such as to reduce the bitrate of the audio-video content.

(105) Referring to FIG. **8**, an exemplary time line is illustrated. A first CSSCI type payload (e.g., CSSCI-0) has a first set of association information P while a second CSSCI type payload (e.g., CSSCI-1) has a second different set of association information P. Having two different association information P for CSSCI-0 and CSSCI-1 distinguish between and identify the two CSSCI payloads. A first time location payload (e.g., Timeline-0) has the first set of association information P that matches the association information P for CSSCI-0, a second time location payload (e.g., Timeline-1) has the same first set of association information P that matches the association information P for CSSCI-0, a third time location payload (e.g., Timeline-2) has the same second set of association information P that matches the association information P for CSSCI-1. In this manner, CSSCI-0, Timeline-0; CSSCI-0, Timeline-1; and CSSCI-1, Timeline-2 are associated together as pairs having spanned watermarked information. This permits the same CSSCI type payload to be used for multiple different time location payloads. As illustrated, two of the temporal location payloads are associated with a previously received CSSCI type payload, and one of the CSSCI type payloads are associated with a subsequently received temporal location payload, and thus bidirectional in its association. In the event that a corresponding CSSCI type payload matching a temporal location payload is not available, then the system may be able to determine that a packet has been lost or otherwise the watermarking was not effective. Similarly, in the event that a corresponding timeline type payload matching a CSSCI payload is not available, then the system may be able to determine that a packet has been lost or otherwise the watermarking was not effective. The loss of watermarking data occurs with some frequency because the audio-video content tends to be modified by audio-video transcoding, such as to reduce the bitrate of the audio-video content.

(106) In an example, a CSSCI type payload (e.g. CSSCI-0) has two sets of association information P0 and P1. A time location payload, e.g. Timeline-0, has two sets of association information P0 and P1 that matches the association information P0 and P1 for CSSCI-0. In this example a bidirectional association exists for the pair CSSCI-0, Timeline-0 where PO points to CSSCI-0 and P1 points to Timeline-0.

(107) The number of bits assigned to the payload identifier (P) may be modified, as desired (e.g., for a desired robustness). Similarly, the number of bits assigned to I, A, T, D, L, and R may be modified, as desired.

(108) In an example, the AV presentation device **180** may maintain a list, which may be denoted by a variable listC for example, of “c” most recently received CSSCI payload(s). “c” may be provided in the watermark, if desired, or otherwise set by the system. In this manner, the AV presentation device **180** may only have to maintain a limited number of CSSCI payloads in memory. In the case that c=1, then once a CSSCI payload is received it remains in effect until another CSSCI payload is received, as illustrated in FIG. **9**. A loss of a CSSCI payload may be detected using the payload

identifier (P); for example, the temporal location payload contains a P that does not correspond to any of the CSSCI payloads in listC. In this manner, the same user experience may be achieved across different AV presentation devices.

(109) In an example, the AV presentation device **180** may maintain more than one list of received CSSCI payload(s). Each list may differ in size and may be maintained (i.e. addition or removal of entries within the list) using a differing set of rules. It is to be understood, that this does not preclude the possibility that a subset of lists may have same size and/or same maintenance rules. As an example, there may be two lists maintained by **180** where one list contains “c1” most recently received CSSCI payload(s) where each payload is received at an interval of “0” CSSCI payload(s); while the other list contains “c2” most recently received CSSCI payload(s), where each payload is received at an interval of “d” CSSCI payload(s).

(110) Referring to FIG. **10**, a modified system may include the content source **100**, the watermark inserter **190**, the MVPD **130**, the broadcast receiving device **160**, and the AV presentation device **180** together with its watermark capable receiver **310** and watermark client **320**. The content server **400** may be modified to include the code database **370**, the metadata server **350**, and one or more of the content and signaling server **380**. The code **300** and the metadata **360** is provided to the content server **400** by the content source **100**. The content and signaling data is provided to the content and signaling server(s) **390**.

(111) The AV presentation device **180** may provide a code in a request based upon the decoded one or more watermarks from the audio-video broadcast. The content server **400** receives the request with the code from the AV presentation device **180**. The metadata server **350** then parses the received code request and based upon information from the code database **370**, makes a request to the content and signaling server(s) **390** to determine the content and signaling information which is then provided to the AV presentation device **180**. In this manner, the AV presentation device **180** only needs to make a single request to a content server **400**, which in turn provides the response to the AV presentation device **180**. It is to be understood that the different functions of the content server **400** may be achieved by combining the existing functions together, separating the existing functions into more components, omitting components, and/or any other technique.

(112) A HTTP or HTTPS request URL (that will be sent to the content server **400**) corresponding to payload(s) in FIG. **5** and FIG. **6**, when time sanative trigger D equals to 1, may be defined as:

(113) TABLE-US-00003 If A is equal to 0 then the HTTP request URL is: HTTP://IIIIIIII.IIIIIIII.IIIIIIII.IIIIIIII/LLLLLLLLL?time=T TTTTTTTTTTTTTTTTTTTTTTT Otherwise, the HTTPS request URL is: HTTPS://IIIIIIII.IIIIIIII.IIIIIIII.IIIIIIII/LLLLLLLLL?time=TTTTTTTTTTTTTTTTTTTTTTT where IIIIIIII.IIIIIIII.IIIIIIII.IIIIIIII above corresponds to the 32-bit IP address signaled in CSSCI payload.

(114) In an example, the subset of URL that specifies information such as: the content server location, the communication protocol, communication port, the login information, the folder on the content server are carried in a designated payload type.

(115) In some implementations a syntax element's value may be derived using a decoding process which may access information spanning multiple payloads. For example, the time code may be fragmented into multiple watermark payloads and then reassembled to construct a complete time code. In an example, the time code may correspond to a temporal location within the AV content. In an example, the time code may correspond to timeline data of the AV content.

(116) For example, the payload size may be 50 bits while the desirable metadata may be 66 bits, thus exceeding the payload size of a single watermark. An example of the desirable metadata may be as follows:

(117) TABLE-US-00004 location of content and server (I) 32 bits (IP address) application layer protocol (A) 1 bit (HTTP or HTTPS) time code (T) 25 bits (for 1 year of uniqueness with a granularity of 1 second) time sensitive trigger (D) 1 bit channel identification (L) 5 bits duration for content server 2 bits requests (R)

(118) Another example of the desirable metadata may be as follows:
(119) TABLE-US-00005 location of content and server (I) 32 bits (IP address) application layer protocol (A) 2 bit (00 = HTTP, 01 = HTTPS, 10 = reserved, 11 = reserved) time code (T) 25 bits (for 1 year of uniqueness with a granularity of 1 second) time sensitive trigger (D) 1 bit channel identification (L) 5 bits duration for content server 2 bits requests (R)
(120) Referring to FIG. **11**, a state transition diagram illustrates one technique to calculate the time code. To obtain a time code synchronization a number of consecutive payloads starting with a payload type “start sync”, is followed by payloads of type “not start sync”, with a total being equal to “r”. By using the total of “r” consecutive payloads, each having some time information contained therein, the time synchronization may be determined by calculating an anchor time. After calculating the anchor time code, the time code may be updated by receiving additional payloads that include partial time code information therein in such a manner that does not require receiving another total of “r” consecutive payloads to determine the next time code. One technique to achieve this time synchronization is to partition the time code in consecutive payloads and an incremental time code in each of the consecutive payloads. When the synchronization is lost, such as by changing the channel, the obtain synchronization process is performed. A video display device when first initialized, or turned on, enters the initial obtaining synchronization state.
(121) Referring to FIG. **12**, an exemplary structure of a watermark payload is illustrated. Z indicates the payload type, where Z equal to 1 indicates the start of the time sync and Z equal to 0 indicates not start of time sync. S indicates the time sync payload bits used in determining absolute time code. M indicates the time sync payloads bits used in maintaining the time code.
(122) By way of example, the AV presentation device **180** may receive n=7 consecutive watermark payloads where the first payload has Z=1 while the rest have Z=0. The bits corresponding to “SSSS” are extracted from (t−n+1).sup.th to t.sup.th watermark payload and concatenated together to obtain a 28 bit representation of the time code “T.sub.t” of a temporal location. The anchor time code “C.sub.t” is also set to “T.sub.t”. “T.sub.t” may be represented as SSSS.sub.z=1, t−n+1 . . . SSSS.sub.z=0,t−1SSSS.sub.z=0,t; “C.sub.t”=“T.sub.t”. In another example, constants may be added (to select a future time) and/or multiplied (to change the granularity) to the derived values. In an example, the derived values are mapped to another value by use of a mapping function.
(123) Once the initialization synchronization is obtained, the anchor time and payload time are updated using each payload. This may be performed, for example, as follows:
T.sub.t=*f*(*C*.sub.t−1,*MMMM*.sub.t)
C.sub.t=*g*(*T*.sub.t)
Where, f represents a mapping function that takes two values as input and outputs one value; g represents a mapping function that takes one value as input and outputs one value; / represents integer division with truncation of the result toward zero, For example, 7/4 and −7/−4 are truncated to 1 and −7/4 and 7/−4 are truncated to −1. In an example:
T.sub.t=*C*.sub.t−1+*MMMM*.sub.t
C.sub.t=T.sub.t
As described above, every “n” payloads the anchor time may also be determined using the bits corresponding to “SSSS”. The anchor time determined using “SSSS” may match the anchor time derivation above and can be used to verify the correctness of the maintained time code.
(124) Since the watermark may span a non-zero time, the temporal location of the time code T.sub.t may be determined by a set of rules, such as for example, T.sub.t may correspond to a time instant at the end of the tth watermark payload.
(125) It is to be understood that multiple syntax elements may be combined to form the code. The code may then be mapped either by the AV presentation device **180** or using another server to different syntax element values. For example, the server information (e.g., location of the content and signaling server(s) and/or application layer protocol, etc.) and time code is combined into a single code. The single code is then mapped to a temporal location in the uncompressed audio-

video stream, and location of the content and signaling server(s). In this manner, a single request may be made to the server for additional information.
(126) A limited number of bits may be used for the time code, in such a manner to permits collisions in the time code. For example, using 20 bits for the timecode allows for at most 12 days of uniqueness at a granularity of 1 second. After 12 days the code space corresponding to the timecode will be reused tending to result in collisions.
(127) In an example the watermark payload may be encapsulated within a SDO private data command as SDO payload using one or more cmdIDs. As an example the watermark payload of FIG. **5** or FIG. **6** maybe encapsulated as SDO payload. A cmdID value 0x05 may refer to a watermark based interactive services trigger or a triggered declarative object (TDO) model. A cmdID value 0x06 may refer to a watermark based interactive services trigger (direct execution model). This facilitates the re-use of existing segmentation and reassembly modules built for trigger transportation. The segmented command may be embedded in watermarks, if desired. The SDO private data may be desired, such as illustrated in FIG. **13**, where the packet is included as part of an SDO_payload( ). In some examples, the watermark payload received in this manner maybe passed to an entity or module in the receiver which handles these defined cmdID types. Then segmentation and reassembly functionality of that module could be reused if watermark payload packet needs to be split into multiple packets—depending upon the selected watermark scheme's capacity in terms of number of bits.
(128) Parameter type T is a 2-bit field that indicates whether the instance of SDO private data, or SDOPrivateData, command is part of a segmented variable length command, and if so, whether the instance is the first, middle, or last segment of the segmented variable length command. In one example, SDOPrivateData is defined by the Consumer Electronics Association (CEA) in Section 7.1.11.2 of “CEA: “Digital Television (DTV) Closed Captioning, CEA-708-E, Consumer Electronics Association, June 2013” (CEA-708), and the type field in the SDO private data command is encoded as specified in Section 7.1.11.2 of CEA-708. pr is a flag that indicates, when set to ‘1’, that the content of the command is asserted to be program related. When the flag is set to ‘0’, the content of the command is not so asserted. Length (L) is an unsigned integer that indicates the number of bytes following the header, in the range of 2 to 27, and is represented in the SDO private data command as the set of bits L.sub.4 through L.sub.0 where L.sub.4 is the most significant and Lo is the least significant. cmdID is a signal that identifies the SDO that has defined the syntax and semantics of the SDO_payload( ) data structure to follow. In an example, cmdID is an 8-bit field. The metadata may be encapsulated within SDO private data as SDO payload using one or more cmdIDs as shown in FIG. **14**.
(129) The payload defined in FIG. **5** and FIG. **6** may be encapsulated within a SDO private data command as SDO payload using one or more cmdIDs. A cmdID value 0x05 and 0x06 may refer to encapsulation of payloads defined in FIG. **5** and FIG. **6** respectively. This facilitates the re-use of existing segmentation and reassembly modules built for trigger transportation. The segmented command may be embedded in watermarks, if desired. The SDO private data may be desired, such as illustrated in FIG. **13**, where the payload packet is included as part of SDO_payload( ).
(130) The payload defined in FIG. **12** may be encapsulated within a SDO private data command as SDO payload using one or more cmdIDs. A cmdID value 0x05 may refer to encapsulation of payload defined in FIG. **12**. This facilitates the re-use of existing segmentation and reassembly modules built for trigger transportation. The segmented command may be embedded in watermarks, if desired. The SDO private data may be desired, such as illustrated in FIG. **13**, where the packet is included as part of SDO_payload( ).
(131) An example of a watermark associated information retrieval system is described next.
(132) The system consists of a watermark detector, an AV presentation device, a watermark information server. In one example, the watermark detector may reside inside an AV presentation device. In one example, the AV presentation device may be a AV presentation device **180**. In one

example, the watermark information server may be an enhanced service information providing server **140**.

(133) In one example, the watermark detector may detect and decode the watermark. The watermark may be an audio watermark. The watermark detector and/or AV presentation device may use the information in the watermark to identify a timeline location of the media content in which the watermark is embedded and/or an address (e.g. IP address) of a server that can be concatenated to obtain further information associated with the watermark. In an example, this may be necessary as the watermark payload capacity may be only a few bits. For example the capacity may be 50 bits over a time period of 1 second or 1.5 seconds or 2 seconds. In this case the AV presentation device may contact a watermark information server to obtain more information about the current timeline location for the current media. The watermark server may send “watermark associated information” as a response to this request.

(134) JavaScript Object Notation (JSON) is a data interchange format.

(135) JSON schema defines a JSON based format for defining the structure of JSON data. JSON schema is intended to define validation, documentation, hyperlink navigation, and interaction control of JSON data.

(136) An object is an unordered collection of zero or more name and value pairs, where a name is a string and a value is a string, number, Boolean, null, object, or array.

(137) A JSON schema is a JSON document, which may be an object. Object properties defined by JSON schema are called keywords or schema keywords.

(138) A JSON schema may contain properties which are not schema keywords.

(139) A JSON value may be an object, array, number, string, or one of false, null, or true.

(140) The terms element and key and keyword and name may be used interchangeably in this document. The term key may be used to refer the name of an object in this document.

(141) The terms recovery file format and recovery data table may be used interchangeably in this document.

(142) FIG. **15** shows an exemplary JSON schema for watermark associated information. With respect to FIG. **15** the following should be noted.

(143) Instead of using XML to represent the watermark associated information retrieved, JSON may be used. In this case instead of using elements (e.g. XML elements) and attributes (XML attributes), JSON objects are used with their properties.

(144) Entertainment identifier register (EIDR) is a universal identifier system for movie and television assets. From top level titles, edits, and DVDs, to encodings, clips and mash-ups, EIDR provides global identifiers for the entire range of audiovisual object types that are relevant to entertainment commerce. EIDR format described at HTTP://eidr.org/documents/EIDR_ID_Format_v1.2.pdf and is incorporated herein by reference. Subsequent versions of EIDR identifier format may be used.

(145) Advertising identifier (AD-ID), which may also be referred to as Ad-ID, is an industry standard for identifying advertising assets across media platforms. Ad-ID code structure as shown in HTTP//www.Ad-ID.org/how-it-works/Ad-ID-structure and is incorporated herein by reference.

(146) With respect to FIG. **15** a regular expression based syntax is defined in JSON schema for EIDR and Ad-ID information inclusion in the ContentID event. In contrast to using a string, this formal syntax enforces that only valid values could be signaled for EIDR and Ad-ID.

(147) This is illustrated in the extracted part of the JSON schema from FIG. **15** below:

(148) TABLE-US-00006 ″ContentID″: { ″type″: ″object″, ″properties″: { ″oneOf″: [{″Type″: {″type″: ″string″, ″enum″: [″EIDR″]}, ″CID″: {″type″: ″string″, ″pattern″: ″{circumflex over ( )}10\\.5240\\/ ([0-9a-fA-F]{4}-){5}[0-9A-Z]$″, ″minLength″: 34, ″maxLength″: 34}}, {″Type″: {″type″: ″string″, ″enum″: [″AD-ID″]}, ″CID″: {″type″: ″string″, ″pattern″: ″{circumflex over ( )}[1-9a-zA-Z]{1} [0-9a-zA-Z]{10} (H|D)?$″, ″minLength″: 11, ″maxLength″: 12}}] } },

(149) In this schema with the use of regular expression based “pattern”, the string included as a value for the content identifier value (CID), or ContentID, key for an EIDR type is by design always a valid EIDR string. In this schema with the use of regular expression based pattern the string included for the CID key for an AD-ID type is by design always a valid Ad-ID string. As a result invalid EIDR and Ad-ID strings cannot be sent from the watermark server in the JSON data.
(150) Further with respect to FIG. **15**, an enumerated data type is defined for the ContentID type instead of a general purpose string. This restricts the value to only valid values. As a result it is not possible to define an invalid value for ContentID type. This can be seen in the use of “enum”: [“EIDR”] and “enum”:[“AD-ID”] values defined for the respective Type (or type) strings inside the ContentID object in the schema in FIG. **15**. As a result invalid ContentID type values could be defined and returned.
(151) Further with respect to FIG. **15**, an extension mechanism is defined for trigger events represented by a trigger key in FIG. **15** to return one or more of a universal resource indicator (URI) event type other than a currently defined URI types in the future. The extended URI types are defined with a designated prefix. This is illustrated in the extracted part of the JSON schema from FIG. **15** below. In an example, the JSON schema includes an application information table (AIT), a media presentation description (MPD), and/or an electronic service guide (ESG) value.
(152) TABLE-US-00007 ″Trigger″: { ″type″: ″object″, ″properties″: { ″Trigger″: {″type″: ″string″, ″format″: ″uri″}, ″Version″: {″type″: ″integer″}, ″UriType″: {″type″: ″string″, ″oneOf″: [{″enum″: [″AIT″, ″MPD″,″ESG″]}, {″pattern″: ″{circumflex over ( )}EXT″}]}, ″required″: [″Trigger″,″UriType″] } },
(153) It can be seen that trigger types, such as AIT, MPD, and ESG, are defined as valid string values. This is represented by the “enum”:[“AIT”, “MPD”,“ESG”] for the UriType string. In the future other valid trigger types may be defined. This is accomplished by allowing use of other string values for UriType. In an example, these strings may start with a prefix of EXT. This behavior is defined by the use of “oneOf” constraint for the UriType string as follows:
(154) TABLE-US-00008 ″UriType″: {″type″: ″string″, ″oneOf″: [{″enum″: [″AIT″, ″MPD″,″ESG″]}, {″pattern″: ″{circumflex over ( )}EXT″}]},
(155) The defined prefix above is denoted as EXT, which means that an extension UriType may start with the characters EXT. Other characters could instead be used. For example instead of EXT, the strings FUT, NEXT or any other suitable string may be used.
(156) In an example, a future UriType may be allowed to be any valid string in which case the relevant part of the schema may be defined as:
(157) TABLE-US-00009 ″UriType″: {″type″: ″string″, ″oneOf″: [{″enum″: [″AIT″, ″MPD″,″ESG″]}, {″pattern″: ″.+″}]},
(158) In another example, additional overall extensions of the schema shown in FIG. **15** and FIGS. **16**A-C are supported for future extensibility. In another example, to allow future extensibility the JSON schema may be defined with key, value pair of additionalProperties: true.
(159) For example the last 4 lines of the JSON schema of may be replaced with following:
(160) TABLE-US-00010 ″required″: [″TimeAnchor″,″IntervalCodeAnchor″,″Event″], ″additionalProperties″: true } }
(161) This allows defining additional objects and types with properties inside the returned JSON data.
(162) FIG. **16**A illustrates logical structure of a JavaScript Object Notation schema. FIG. **16**B illustrates the left half part of FIG. **16**A. FIG. **16**C illustrates the lower half part of FIG. **16**A. The FIGS. **16**A-C structure corresponds to FIG. **15** JSON schema. However some or part of the logical structure may be manifested with variant JSON schema.
(163) In an example, the watermark associated information returned via JSON schema illustrated in FIG.**15** and/or FIGS. **16**A-C from a watermark server may be a recovery file format.

(164) In another example, instead of using JSON to represent the watermark associated information returned from the watermark server XML format may be used for it. A few enhancements and ways of returning XML format data from watermark server and conformance with defined XML schema is described next.
(165) In an example, a pattern based syntax using XML is defined for one or more of a EIDR and an Ad-Id information inclusion in the ContentID event. In contrast to using a general purpose xs:String data type, this formal syntax enforces that only valid values could be signaled for EIDR and AD-ID.
(166) In an example, the XML schema for EIDR information inclusion is:
(167) TABLE-US-00011 <xs:simpleType name=″CID″> <xs:restriction base=″xs.token″> <xs:pattern value=″(10\.5240/([0-9a-fA-F]{4}-){5}[0-9A-Z])″/> </xs:restriction> </xs:simpleType>
(168) In an example, the XML schema for Ad-ID information inclusion is:
(169) TABLE-US-00012 <xs:simpleType name=″CID″> <xs:restriction base=″xs:token″> <xs:pattern value=″([1-9a-zA-Z]{1}[0-9a-zA-Z]{10}(H|D)?)″/> </xs:restriction> </xs:simpleType>
(170) A combined XML schema for EIDR or Ad-ID inclusion is as shown below:
(171) TABLE-US-00013 <xs:simpleType name=″CID″> <xs:restriction base=″xs:token″> <xs:pattern value=″([1-9a-zA-Z]{1}[0-9a-zA- Z]{10}(H|D)?)|(10\.5240/([0-9a-fA-F]{4}-){5}[0-9A-Z])″/> </xs:restriction> </xs:simpleType>
(172) In another example, an enumerated data type is defined for the type of ContentID, or ContentIDType, instead of a general purpose string. This restricts the value to only valid values.
(173) In an example, the XML schema for this is:
(174) TABLE-US-00014 <xs:simpleType name=“ContentIDType”> <xs:restriction base=“xs.string”> <xs.enumeration value=“EIDR” /> <xs.enumeration value=“Ad-ID” /> <xs.enumeration value=“EXT” /> </xs.restriction> </xs:simpleType>
(175) The overall XML schema for ContentID event with inclusion of constrained type and contentID attributes is as shown below:
(176) TABLE-US-00015 <xs.element name=“ContentID”> <xs.complexType> <xs.attribute name=“type” type=“ContentIDType”/> <xs.attribute name=“contentID” type=“CID”/> </xs:complexType> </xs:element
(177) Another example of using XML schema is described next.
(178) The above XML schema definitions allows defining a ContentID event with type equal to Ad-ID but the ContentID value defined for EIDR identifier.
(179) Similarly the above XML schema definitions allow defining a ContentID event with type equal to EIDR but the ContentID value defined for Ad-ID identifier.
(180) To prevent this definition, the XML schema may be defined as follows.
(181) TABLE-US-00016 <xs:simpleType name=“ContentIDType1”> <xs:restriction base=“xs:string”> <xs:enumeration value=“EIDR” /> </xs:restriction> </xs:simpleType> <xs:simpleType name=“ContentIDType2”> <xs:restriction base=“xs:string”> <xs:enumeration value=“Ad-ID”/> </xs:restriction> </xs:simpleType> <xs:simpleType name=“CIDToken1”> <xs:restriction base=“xs:token”> <xs:pattern value=“10\.5240/([0-9a-fA-F]{4}-){5}[0-9A-Z]”/> </xs:restriction> </xs:simpleType> <xs:simpleType name=“CIDToken2”> <xs:restriction base=“xs:token”> <xs:pattern value=“[1-9a-zA-Z]{1}[0-9a-zA-Z]{10}(H|D)?”/> </xs:restriction> </xs:simpleType> <xs:element name=“ContentID”> <xs:complexType> <xs:choice> <xs:sequence> <xs:element name=“CID1”> <xs:complexType> <xs:attribute name=“type” type=“ContentIDType1”/> <xs:attribute name=“contentID” type=“CIDToken1”/> </xs:complexType> </xs:element> </xs:sequence>

<xs:sequence> <xs:element name="CID2"> <xs:complexType>
<xs:attribute name="type" type="ContentIDType2"/>
<xs:attribute name="contentID" type="CIDToken2"/> </xs:complexType>
</xs:element> </xs:sequence> </xs:choice>
</xs:complexType> </xs:element>

(182) This XML schema strictly enforces that ContentID value can only be a valid Ad-ID identifier value when ContentID event has type equal to Ad-ID,. Also the XML schema enforces that ContentID value can only be a valid EIDR identifier value when ContentID event has type equal to EIDR.

(183) Cardinality of query spread attribute is modified from 0 . . . N to 0 . . . 1. Signaling multiple query spread values can result in confusing receiver behavior thus at most only 1 value of query spread is signaled.

(184) In an example, the XML schema for this is:

(185) TABLE-US-00017 <xs:element name="RecoveryData"> <xs:complexType>
<xs:attribute name="querySpread" type="xs:string" use="optional"/>
</xs:complexType> </xs:element>

(186) An extension element is defined for future extensions. A RecoveryExt element of the type "RecoveryExtType" is defined to allow defining proprietary extensions of the watermark associated information.

(187) In an example, the following XML schema may be defined for this.

(188) The extension element may be defined as a child element of the overall root element or at any other suitable place in the overall XML schema as follows:

(189) TABLE-US-00018 <xs:element name="RecoveryExt" type="RecoveryExtType" minOccurs="0"/>

(190) In an example, the XML data type for the extension element RecoveryExtType is:

(191) TABLE-US-00019 <xs:complexType name="RecoveryExtType">
<xs:sequence> <xs:annotation> <xs:documentation>
Proprietary extensions of recovery file format. It is a requirement that different namespace may be used for proprietary extensions.

(192) TABLE-US-00020 </xs:documentation>
</xs:annotation> <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence> </xs:complexType>

(193) Additional examples are now described for JSON schema to make the schema extensible.

(194) An additional JSON schema for watermark associated information retrieval is shown in FIG. **17**. In the schema support is included for extensibility of JSON schema for extensibility.

(195) In the JSON schema in FIG. **17** compared to JSON schema in FIG. **15**, the following is included for extensibility: (1) A top level key, such as RecoveryFF is defined and the current recovery file format schema in FIG. **17** is defined as an object for this top level key. Thus, the schema in FIG. **15** may be wrapped inside the top level key. This allows extensibility for the current recovery file format by use of "allOf" and "$ref" keywords of a JSON schema.

(196) The JSON keyword "allOf" is defined in HTTP://tools.ietforg/html/draft-fge-json-Schema-validation-00 which is incorporated here by reference. In an example, "allOf" defines that the given data may be valid against all schema defined by a keyword value. In an example, to validate against "allOf", the given data may be valid against all schemas defined by this keyword's value.

(197) Part of a schema may be referred to using the $ref keyword. $ref gets logically replaced with the schema part that it points to.

(198) (2) Keys are made unique. Thus none of the keys, even if they belong to different objects, have the same key name. This facilitates extensibility.

(199) One example of doing an extension is when a second version of a recovery file format is

defined. In this case the new key and value pairs may be added to a schema, while keeping the old key value pairs for backward compatibility with the schema in FIG. **17**.
(200) When extending the JSON schema in FIG. **17**, the following type of new key may be defined:
(201) TABLE-US-00021 “RecoveryFFV2”: { “type”: “object”, “properties”: { “V2”: {“allOf”: [{“$ref”: “#/RecoveryFF”}, { “properties”: { “newkeyA”: “valueA”, “newkeyB”: “valueB” }, “required”: [“newkeyA”] }]} } }
(202) In an example, the new key is the “RecoveryFFV2” key.
(203) In this case with the use of allOf keyword the data may be valid against all of the given schemas.
(204) The first schema included inside allOf keyword is {“$ref”:“#/RecoveryFF”}, which refers to the schema for first version of recovery file format as shown in FIG. **17**. The new keys and values for schema for the second version of recovery file format are then included as:
(205) TABLE-US-00022 { “properties”: { “newkeyA”: “valueA”, “newkeyB”: “valueB” }, “required”: [“newkeyA”] }
(206) Thus the overall example schema for second version of recovery file format is as shown in FIG. **18**.
(207) A second example for providing an extensible schema is illustrated below. In this example, a JSON for Linked Data (JSON-LD) based extension mechanism is defined by inclusion of a @context keyword. JSON-LD is a format to serialize linked data. The JSON-LD syntax is designed to integrate into systems that already use JSON and provides an upgrade path from JSON to JSON-LD. JSON-LD is primarily intended to be a way to use linked data in web-based programming environments, to build interoperable web services, and to store linked data in JSON based storage engines. Linked data is a way to create a network of standards-based machine interpretable data across different documents and web sites. It allows an application to start at one piece of linked data and follow embedded links to other pieces of linked data that are hosted on different sites.
(208) FIG. **19** shows an example of a JSON schema for a watermark associated information retrieval.
(209) In the JSON schema in FIG. **19**, compared to the JSON schema in FIG. **15**, the following is added for extensibility:
(210) (1) A key (@context) is defined and the current recovery file format schema in FIG. **19** is included as “RecoveryFF”: HTTP://www.atsc.org/contexts/3.0/RecoveryFF inside the key (@context). The schema is then wrapped inside the key “RecoveryFF”.
(211) (2) Keys are made unique, in that keys do not have the same key name, even if they below to different objects. This facilitates extensibility.
(212) One example of enabling an extension is defining a second version of the recovery file format. In this example, the new key and value pairs will be added to a schema, while keeping the old key value pairs for backward compatibility with the schema in FIG. **19**.
(213) When enabling an extension of the JSON schema in FIG. **19**, a @context can be included for the new keys and values as:
(214) TABLE-US-00023 “@context”: { “RecoveryFF2”: “HTTP://www.atsc.org/contexts/3.1/RecoveryFF” }, “RecoveryFF2”: { “type”: “object”, “properties”: { “newkeyA”: “valueA”, “newkeyB”: “valueB” }, “required”: [“newkeyA”] }
(215) The new key in this case is “RecoveryFF2”.
(216) The new “@context” for the new key “RecoveryFF2” is “RecoveryFF2” “HTTP://www.atsc.org/contexts/3.1/RecoveryFF”
(217) In an example, the new keys and values for the schema for the second version of recovery file format are then:

(218) TABLE-US-00024 “RecoveryFF2”: { “type”: “object”, “properties”: { “newkeyA”: “valueA”, “newkeyB”: “valueB” }, “required”: [“newkeyA”] }

(219) The overall example schema for second version of recovery file format is shown in FIG. **20**.

(220) In an example, when doing an extension a second version of recovery file format is defined, only one new @context may be included for old and new keys and values as follows.

(221) TABLE-US-00025 “@context”: { “RecoveryFF”: “HTTP://www.atsc.org/contexts/3.1/RecoveryFF” },

(222) In this example, the new keys and values for this second version of recovery file format are: “newkeyA”:“valueA”, “newkeyB”:“valueB”,
which may be included with other keys and values in the previous version of the schema in FIG. **19**.

(223) Thus, in this example, by changing the value of the “RecoveryFF” key inside “@context”, the new and old keys and values may be included together as shown in FIG. **21**.

(224) Thus the overall schema for a recovery file format is as shown in FIG. **21**.

(225) Alternative examples are now provided for recovery file format structure.

(226) An alternative logical structure of recovery file format table is shown in FIG.**22**. The logical structure of Component Description is shown in FIG. **23**. Three different logical structures of Component Anchor are shown in FIG. **24**A, FIG. **24**B, and FIG. **24**C. With respect to FIG. **22** semantics of various elements are as described below.

(227) ThisComponent—A description of the media component embedded with a watermark.

(228) serverCode—When present, this element may provide the serverCode value which was employed in the HTTP request to which this recovery file was provided as a response.

(229) intervalCode—When present, this element may provide the intervalCode value from the query request to which the recovery data table was provided as a response.

(230) ComponentDescription—A data element describing ThisComponent in the format defined in FIG. **23**.

(231) querySpread—When present, this element may express the maximum duration that the receiver is recommended to delay submission of a HTTP request.

(232) OtherComponent—An element describing another watermarked media component associated with the same service as ThisComponent in the format defined in FIG. **23**.

(233) ContentID—This field may identify a content identifier.

(234) Instead of using a ContentID List container object, an array of ContentID objects is defined with effective cardinality of 0 . . . N instead of 1 . . . N (i.e., 0 to N entries in the array). This allows easier parsing of JSON data and simplifies overall data structure by not requiring a container object, which adds more parsing complexity, while still maintaining the desired flexibility.

(235) ContentID.type—A field that is preferably required when ContentId element is included. Two values are may be defined:

(236) “EIDR” indicates a content identification per the EIDR registry.

(237) “Ad-ID” indicates a content identifier per the Ad-ID registry.

(238) ContentID.cid—A field that is used when ContentId element is included that provides the content identification. The type of content identifier may be as given in the ContentID.type attribute. Either an EIDR (34-character canonical form with hyphens) or Ad-ID (11 or 12-character canonical form) can be included.

(239) ContentID.validFrom—A field that provides information about when the ContentId is valid from.

(240) ContentID.validUntil—A field that provides information about up to when the ContentId is valid until.

(241) SourceID—An element describing a distribution source that employs ATSC emission specifications. This element is applicable to circumstances where the watermarked content is

included in the redistribution of a service that is broadcast in accordance with ATSC emission specifications.
(242) country—Country code associated with the primary administrative entity under which the value provided in bsid is assigned, using the applicable alpha-2 country code format as defined in ISO 3166-1. ISO 3166-1 available at http://webstore.ansi.org/RecordDetail.aspx?sku=ISO %203166-1:2013 is incorporated here by reference. bsid—The Broadcast service identifier (BSID) of the ATSC distribution source.
(243) majorChannelNo—The major channel number assigned to the ATSC distribution source. This value is scoped to the BSID.
(244) minorChannelNo—The minor channel number assigned to the ATSC distribution source. This value is scoped to the BSID.
(245) Service—This element describes the service, its signaling formats and broadband locations.
(246) serviceId—16-bit integer that may uniquely identify this Service within the scope of this Broadcast area.
(247) sltSvcSeqNum—This integer number may indicate the sequence number of the service information with service identifier equal to the serviceId attribute above. sltSvcSeqNum value may start at 0 for each service and may be incremented by 1 every time any attribute or child of this Service element is changed. If no attribute or child element values are changed compared to the previous Service element with a particular value of serviceID then sltSvcSeqNum may not be incremented. The sltSvcSeqNum field may wrap back to 0 after reaching the maximum value.
(248) slsProtocol—specifies the signaling format associated with this service.
(249) slsMajorProtcolVersion—Major version number for the signaling protocol specified in slsProtocol.
(250) slsMinorProtocolVersion—Minor version number for the signaling protocol specified in slsProtocol.
(251) svcInetUrl—Provides information about URL to access ESG or service level signaling files for this service via broadband, if available.
(252) URLtype—Type of files available with svcInetUrl.
(253) URLValue—URL to access Internet signaling files for this service identified by serviceidentifier serviceId.
(254) The URL value property (URLValue) is defined for indicating service internet URL value inside a contained object which encompasses the service Internet URL related properties. URLtype may be a required property (instead of optional property) for service Internet URL because otherwise it will not be known what type of URL is signaled.
(255) With respect to FIG. **23** semantics of various elements are as described below.
(256) ComponentDescription—Provides a description of a watermarked media component associated with a service.
(257) ComponentAnchor—Information about the first payload in the watermarked media component as defined in either FIG. **24**A or FIG. **24**B or FIG. **24**C.
(258) mediaType—A string with value “audio” to indicate that the description applies to an audio component only, “video” to indicate that the description applies to a video component only, or “both” to indicate that the description applies to both an audio and video component.
(259) descriptor—An arbitrary descriptive string associated with the watermarked media component intended for consumption by an application.
(260) priority—A numeric value indicating the relative priority of the described component. When no priority value is indicated for a component, its priority may be 0.
(261) With respect to FIG. **24**A semantics of various elements are as described below.
(262) ComponentAnchor—An element that specifies characteristics of the first payload in a video or audio watermark segment.
(263) intervalCodeAnchor—The intervalCode in the first payload in a video or audio watermark

segment.
(264) PresentationTime—The wall clock presentation time of the first frame of the first message block in the video watermark segment, or, for audio components, the wall clock presentation time of the first sample of the first symbol in the first cell of the audio watermark segment.
(265) JSON schema for recovery file format with logical structure shown in FIG. **22**, FIG. **23**, FIG. **24**A is shown below as FIG. **25**A-D.
(266) With respect to JSON schema, in an alternative example the presentationTime may be signaled with a data type other than the type “string”. For example type “number” may be used.
(267) In this case with respect to FIG. **24** the corresponding JSON schema part would be as follows:
(268) “presentationTime”: {“type”:“number”} instead of as ““presentationTime”: {“type”:“string”}
(269) In an alternative example the presentationTime may be signaled with a data type other than the type “string”. For example type “integer” may be used.
(270) In this case with respect to FIG. **24** the corresponding JSON schema part would be as follows:
(271) “presentationTime”: {“type”:“integer”} instead of as ““presentationTime”: {“type”:“string”}
(272) With respect to recovery file format logical structure, in an alternative example a parent element ATSCSourceID and additionally a choice selection may be used inside the container SourceID element. This may allow defining source identifier other than ATSC in future. This part of the recovery file format logical structure may be as shown below.
(273) TABLE-US-00026 SourceID . . . 1 Choice ATSCSourceID country tring ISO 3166-1 alpha-2 country code associated with the primary administrative entity under which the given bsid is assigned. bsid nteger Identifier of the whole Broadcast Stream. The value of BSID may be unique on a regional level (for example, North America). An administrative or regulatory authority may play a role. majorChannelNo nteger An integer number in the range 1 to 1000 representing the “major” channel number of the service. minorChannelNo nteger An integer number in the range 1 to 1000 representing the “minor” channel number of the service.
(274) In this case the semantics of SourceID and ATSCSourceID may be as follows: SourceID—An element describing a distribution source to which the watermarked content is attributed.
(275) ATSCSourceID—An element describing a distribution source that employs ATSC emission specifications. This element is applicable to circumstances where the watermarked content is included in the redistribution of a service that is broadcast in accordance with ATSC emission specifications.
(276) In this case the part of JSON schema corresponding to this may be as shown below.
(277) TABLE-US-00027 “SourceID”: { “type”: “object”, “properties”: { “oneOf”: [{ “ATSCSourceID”: {“type”: “object”, “properties”: { “country”: {“type”: “string”,“pattern”: “{circumflex over ( )}[a-zA-Z]{2}$”}, “bsid”: {“type”: “integer”,“minimum”: 0, “maximum”: 65535 }, “majorChannelNo”:{“type”: “integer”,“minimum”: 1, “maximum”: 999 }, “minorChannelNo”:{“type”: “integer”,“minimum”: 1, “maximum”: 999 } }, “required”: [“country”,“bsid”,“majorChannelNo”,“minorChannelNo”] }]}},
(278) Alternative example logical structure for component anchor is shown in FIG. **24**B. The main difference between FIG. **24**B and FIG. **24**A is that instead of defining a single presentationTime element or key or property to represent presentation time, two elements or keys or properties presentationTime and presentationTimeMsec are defined.
(279) With respect to FIG. **24**B semantics of various elements are as described below.
(280) ComponentAnchor—An element that specifies characteristics of the first payload in a video or audio watermark segment.

(281) intervalCodeAnchor—The intervalCode in the first payload in a video or audio watermark segment.
(282) PresentationTimeInteger—The integer part—first 32 bits of 64-bit Network Time Protocol (NTP) formatted wall clock presentation time of the first frame of the first message block in the video watermark segment, or, for audio components, the wall clock presentation time of the first sample of the first symbol in the first cell of the audio watermark segment.
(283) PresentationTimeFraction—The fraction part—last 32 bits of 64-bit NTP formatted wall clock presentation time of the first frame of the first message block in the video watermark segment, or, for audio components, the wall clock presentation time of the first sample of the first symbol in the first cell of the audio watermark segment. In this 32-bit fraction part non-significant low-order bits may be set to 0.
(284) JSON schema for recovery file format with logical structure shown in FIG. **22**, FIG. **23**, FIG. **24**B is shown as FIG. **26**A-D.
(285) Alternative example logical structure for component anchor is shown in FIG. **24**C. The main difference between FIG. **24**C and FIG. **24**A is that instead of defining a single presentationTime element or key or property to represent presentation time, two elements or keys or properties presentationTime and presentationTimeMsec are defined.
(286) With respect to FIG. **24**C semantics of various elements are as described below.
(287) ComponentAnchor—An element that specifies characteristics of the first payload in a video or audio watermark segment. intervalCodeAnchor—The intervalCode in the first payload in a video or audio watermark segment.
(288) PresentationTime—This 32-bit unsigned integer may indicate the presentation time of the first frame of the first Message block in the video watermark segment, or, for audio components, as the least-significant 32 bits of the count of the number of seconds since Jan. 1, 1970 00:00:00, International Atomic Time (TAI).
(289) PresentationTimeMsec—This 10-bit unsigned integer in the range 0 to 999 may indicate the milliseconds offset from the time indicated in PresentationTime, such that the formula PresentationTime+(PresentationTimeMsec/1000) yields the actual presentation time of the first frame of the first Message block in the video watermark segment, or, for audio components to the nearest 1 millisecond. (PresentationTimeMsec/1000) mean PresentationTimeMsec divided by 1000.
(290) JSON schema for recovery file format with logical structure shown in FIG. **22**, FIG. **23**, FIG. **24**C is shown as FIG. **27**A-D.
(291) An alternative logical structure of recovery file format table is shown in FIG.**28**. The logical structure of component description is shown in FIG. **29**. The logical structures of component anchor is shown in FIG. **30**. FIG. **31** illustrates exemplary slsProtocol values for recovery file format in FIG. **28**. FIG. **32** illustrates exemplary urlType values for recovery file format in FIG. **28**.
(292) With respect to FIG. **28** semantics of various elements are as described below. thisComponent—A description of the media component embedded with a watermark containing the VP**1** payload containing serverCode and intervalCode. VP**1** payload is specific arrangement of the 50-bit audio watermark payload data. serverCode—When present, this element may provide the serverCode value employed in the HTTP request to which this recovery file was provided as a response. intervalCode—When present, this element may provide the intervalCode value from the query request to which the recovery data table was provided as a response. componentDescription—A data element describing thisComponent in the format defined in FIG. **29** and the parameter descriptions that follow. querySpread—When present, this element may express the maximum duration that the receiver is recommended to delay submission of a dynamic event HTTP request, in units of 1 millisecond. The expectation is that the receiver will apply a small enough level of granularity to achieve an even spread of queries across the querySpread duration, such as 1 millisecond. otherComponent—An element describing another watermarked media component

associated with the same service as thisComponent in the format defined in FIG. **25**A-D and the parameter descriptions that follow. contentID—This field may identify a content identifier. contentID.type—Type of Content ID system used for this Content ID. Three Values are defined currently by ATSC: “EIDR” indicates a content identification per the EIDR registry as defined in (http://eidr.org). “Ad-ID” indicates a content identifier per the Ad-ID registry as defined in (http://ad-id.org).
(293) “UserPriv” indicates a user private content identifier
(294) Additional Content ID system types may be defined by ATSC in the future.
(295) For “UserPriv” content identifier, care should be taken that the contentID.cid is unique among Content ID system types that appear in this broadcast stream.
(296) An alternative semantics for contentID.type may be as follows:
(297) Type of Content ID system used for this Content ID. Two values may be, for example, defined by ATSC: “EIDR” indicates a content identification per the EIDR registry (http://eidr.org). “Ad-ID” indicates a content identifier per the Ad-ID registry (http://ad-id.org).
(298) Additional types for user private content identifiers can be defined. These may use a prefix of “x-” to indicate a user private content identifier type.
(299) Additional Content ID system types may be defined by ATSC.
(300) For user private content identifier types, the contentID.type for such systems preferably does not duplicate any Content ID system type defined by ATSC and is unique among Content ID system types that appear in the broadcast stream.
(301) It should be noted that Instead of requiring usage of “x-” as prefix for user private content identifiers, any other specified prefix may be used. For example prefix of “UserPriv-” may be specified to be used.
(302) Also instead of the “user private content identifiers” the term “private use content identifiers” may be used. contentID.cid—A field that is required when contentID element is included that provides the content identification. In the case of the EIDR Content ID system, this may be the 34-character canonical form (with hyphens) of the identifier. In the case of the Ad-ID Content ID system, this may be the 11-character or 12-character canonical form of the identifier. In the case of a UserPriv Content ID system (e.g. House Numbers, ISCI, etc.), the format of the identifier is determined by the specification of the system.
(303) House Number may include broadcaster specific private content identifiers. For example Broadcasting Company A may use their private content identifiers as their private House Numbers and Broadcasting Company B may use their private content identifiers as their private House Numbers.
(304) Industry Standard Coding Identification (ISCI) was a standard created to identify commercials that aired on TV in the United States, for ad agencies and advertisers from 1970 until 2004. It was replaced by Ad-ID in 2005.
(305) An alternative semantics for contentID.value may be as follows: contentID.cid—A field that is used when contentID element is included that provides the content identification. In the case of the EIDR Content ID system, this may be the 34-character canonical form (with hyphens) of the identifier. In the case of the Ad-ID Content ID system, this may be the 11-character or 12-character canonical form of the identifier. In the case of a user defined Content ID type (with prefix “x-” for contentID.type) or any other ATSC private Content ID system (e.g. House Numbers), the format of the identifier may be determined by the specification of the system. contentID.validFrom—Start time of the interval of validity of the contentID value. contentID.validUntil—End time of the interval of validity of the contentID value. sourceID—An element describing an attributable distribution source that employs ATSC emission specifications. This element is applicable to circumstances where the watermarked content is included in the redistribution of a service that is broadcast in accordance with ATSC specifications. country—Country code associated with the primary administrative entity under which the value provided in bsid field below is assigned, using

the applicable alpha-2 country code format as defined in ISO 3166-1. ISO 3166-1 is defined in ISO: ISO 3166-1:2013 (E/F), “Codes for the representation of names of countries and their subdivisions—Part 1: Country codes,” International Organization for Standardization, 3rd Edition, Nov. 11, 2013, and incorporated by reference here in its entirety. bsid—The BSID of the attributable ATSC distribution source. majorChannelNo—The major channel number assigned to the attributable ATSC distribution source. This value is scoped to the BSID. minorChannelNo—The minor channel number assigned to the attributable ATSC distribution source. This value is scoped to the BSID. service—This element describes the service, its signaling formats and broadband locations. serviceId—16-bit integer that may uniquely identify this Service within the scope of this Broadcast area. sltSvcSeqNum—This integer number may indicate the sequence number of the Service List Table (SLT) service information with service ID equal to the serviceId attribute above. sltSvcSeqNum value may start at 0 for each service and may be incremented by 1 every time any attribute or child of this service element is changed. If no attribute or child element values are changed compared to the previous service element with a particular value of serviceID then sltSvcSeqNum may not be incremented. The sltSvcSeqNum field may wrap back to 0 after reaching the maximum value.

(306) SLT is a table of signaling information which is used to build a basic service listing and provide discovery for signaling which provides information for acquisition of ATSC 3.0 services and their content components. slsProtocol—Specifies the signaling format associated with this service, with permitted values and their meanings as shown in FIG. **31**. slsMajorProtcolVersion—Major version number for the signaling protocol specified in slsProtocol. slsMinorProtocolVersion—Minor version number for the signaling protocol specified in slsProtocol. svcInetUrl—Base URL to access Electronic Service Guide (ESG) or service level signaling files for this service via broadband, if available. urlType—Type of files available with svcInetUrl, with permitted values and their meaning as shown in FIG. **32**. urlValue—URL to access Internet signaling files for this service identified by service identifier serviceId.

(307) An exemplary JSON schema for recovery file format with logical structure shown in FIG. **28**, is shown as FIG. **33**A-E.

(308) Another exemplary JSON schema for recovery file format with logical structure shown in FIG. **28**, is shown as FIG. **34**A-D.

(309) Another exemplary JSON schema for recovery file format with logical structure shown in FIG. **28**, is shown as FIG. **35**A-E.

(310) The schema in FIG. **35**A-E allow any “string” data type value to be used for “type” field in one of the cases. This allows defining additional ATSC defined content ID types and user private content ID types.

(311) In yet another example the requirement about using prefix of “x-” for user private content identifier may be enforced in the schema by defining a data type as follows:

(312) TABLE-US-00028 “cid”: { “type”: “string”, “pattern”: “{circumflex over ( )}x-”, },

(313) A difference between JSON schema in FIG. **33**A-E and FIG. **34**A-D relates to a difference in semantics for contentID.type and contentID.cid properties. For schema in FIG. **34**A-D the semantics for the properties contentID.type and contentID.cid may be as follows: contentID.type—Type of Content ID system used for this Content ID. Following values are defined currently by ATSC:

“EIDR” indicates a content identification per the EIDR registry as defined in (http://eidr.org).

“Ad-ID” indicates a content identifier per the Ad-ID registry as defined in (http://ad-id.org).

(314) Additional Content ID system types may be defined by ATSC in the future.

(315) A suitable abbreviation for user defined Content ID system types be used (or may appear here). When this is done, care should be taken that the abbreviation is unique among Content ID system types that appear in this broadcast stream.

(316) contentID.cid-Content ID value. In the case of the EIDR Content ID system, this may be the 34-character canonical form of the identifier. In the case of the Ad-ID Content ID system, this may be the 11-character or 12-character form of the IDENTIFIER. In the case of a user private Content ID system (e.g. House Numbers, ISCI, etc.), the format of the identifier is determined by the specification of the system.
(317) The JSON schemas in FIG. **33**A-E and FIG. **34**A-D include the following as part of schema for the object contentID:
(318) TABLE-US-00029 “contentID”: { “type”: “array”, “items”: { “type”: “object”, “properties”: {“oneOf”: [{ ... }, { ... }, ... {“type”: “object”}]} }, “minItems”: 0 },
where “ . . . ” indicates different JSON schema parts as shown in FIG. **33**A-E and FIG. **34**A-D which are omitted here.
(319) In this case including {“type”:“object”} as part of the schema provides future extensibility. Including {“type”:“object”} as part of schema for contentID object allows in addition to specifically defined constrained schema syntax for contentID which must obey the schema properties of EIDR or AD-ID or “userPriv”, a free form JSON object for contentID object. The syntax of this object can be defined in future. Thus this supports future extensibility. Current version of receivers can skip past a generic JSON object in this case. Also in another example of usage, schema in FIG. **34**A-D including {“type”:“object”} inside contentID object can serve as another way of including any different private content id values.
(320) FIG. **36**A-B illustrate an exemplary recovery file format logical structure. One difference between FIG. **28** and FIG. **36**A-B is that a compact Ad-ID form is supported. In this form Ad-ID may be represented as the ASCII characters representing the decimal value of the 4-byte unsigned integer compact Ad-ID Identifier.
(321) Another exemplary JSON schema for recovery file format with logical structure shown in FIG. **36**A-B, is shown as FIG. **37**A-E.
(322) The schema in FIG. **37**A-E allows an additional pattern entry for Ad-ID type of content identifier, which allows Ad-ID to be represented as the ASCII characters representing the decimal value of the 4-byte unsigned integer Compact Ad-ID. In an example, this is accomplished by defining a pattern that is created by performing a logical OR operation on each individual pattern. In one example this part of the schema may be as shown below.
(323) TABLE-US-00030 “cid”: { “type”: “string”, “pattern”: “{circumflex over ( )}[1-9a-zA-Z]{1}[0-9a-zA-Z]{10}(H|D)? |{circumflex over ( )}[0-9]{1,10}$”, “maxLength”: 12 },
(324) In the examples with respect to FIG. **17**, FIG. **18**, FIG. **19**, FIG. **20**, FIG. **21**, FIG. **25**A-D, FIG. **26**A-D, FIG. **27**A-D, FIG. **28**, FIG. **29**, FIG. **30**, FIG. **31**, FIG. **32** further extensibility the JSON schema may be defined with additionalProperties: true
(325) In other examples that prevent changing the schema with respect to FIG. **17**, FIG. **18**, FIG. **19**, FIG. **20**, FIG. **21**, FIG. **25**A-D, FIG. **26**A-D, FIG. **27**A-D, FIG. **28**, FIG. **29**, FIG. **30**, FIG. **31**, FIG. **32** further extensibility the JSON schema may be defined with additionalProperties: false.
(326) With respect to various JSON schema, in an alternative example data type of some of the element or key or property may be different than those indicated in the schema. For example instead of data type “string” a data type “integer” or “number” or “boolean” or “array” or “object” may be used. Similarly any other JSON date type may instead be signaled as any other JSON data type. All these variations are anticipated in combination with the examples in the detailed description.
(327) Moreover, each functional block or various features of the base station device and the terminal device used in each of the aforementioned embodiments may be implemented or executed by a circuitry, which is typically an integrated circuit or a plurality of integrated circuits. The circuitry designed to execute the functions described in the present specification may comprise a

general-purpose processor, a digital signal processor (DSP), an application specific or general application integrated circuit (ASIC), a field programmable gate array signal (FPGA), or other programmable logic devices, discrete gates or transistor logic, or a discrete hardware component, or a combination thereof. The general-purpose processor may be a microprocessor, or alternatively, the processor may be a conventional processor, a controller, a microcontroller or a state machine. The general-purpose processor or each circuit described above may be configured by a digital circuit or may be configured by an analogue circuit. Further, when a technology of making into an integrated circuit superseding integrated circuits at the present time appears due to advancement of a semiconductor technology, the integrated circuit by this technology is also able to be used.
(328) It is to be understood that the claims are not limited to the precise configuration and components illustrated above. Various modifications, changes and variations may be made in the arrangement, operation and details of the systems, methods, and apparatus described herein without departing from the scope of the claims.

Claims

1. A receiving device for receiving recovery file information, the receiving device comprising: a processor, and a memory associated with the processor; wherein the processor is configured to: receive a recovery data table, wherein the recovery data table includes a contentID element identifying a content identifier and having a cardinality of **0**. . . . N, wherein the contentID element includes: (i) a type element, wherein the type element is defined by two values, which include (a) a first value indicating a content identification per an EIDR registry and (b) a second value indicating a content identifier per an Ad-ID registry, and additional types for user private content identifiers may be used by setting the type element to a third value, and (ii) a cid element, wherein the cid element is a 34-character canonical form with hyphens in a case of an EIDR content ID system, the cid element is 11-character or 12-character form or ASCII characters representing a decimal value of a 4-byte unsigned integer Compact Ad-ID Identifier in a case of an Ad-ID Content ID system, and decode the cid element of the recovery data table.
2. A signaling device for signaling recovery file information, the signaling device comprising: a processor, and a memory associated with the processor; wherein the processor is configured to: encode a recovery data table, wherein the recovery data table includes a contentID element identifying a content identifier and having a cardinality of 0 . . . . N, wherein the contentID element includes: (i) a type element, wherein the type element is defined by two values, which include (a) a first value indicating a content identification per an EIDR registry and (b) a second value indicating a content identifier per an Ad-ID registry, and additional types for user private content identifiers may be used by setting the type element to a third value, and (ii) a cid element, wherein the cid element is a 34-character canonical form with hyphens in a case of an EIDR content ID system, and the cid element is 11-character or 12-character form or ASCII characters representing a decimal value of a 4-byte unsigned integer Compact Ad-ID Identifier in a case of an Ad-ID Content ID system.
3. A method for receiving recovery file information, the method including: receiving a recovery data table, wherein the recovery data table includes a contentID element identifying a content identifier and having a cardinality of 0 . . . . N, wherein the contentID element includes: (i) a type element, wherein the type element is defined by two values, which include (a) a first value indicating a content identification per an EIDR registry and (b) a second value indicating a content identifier per an Ad-ID registry, and additional types for user private content identifiers may be used by setting the type element to a third value, and (ii) a cid element, wherein the cid element is a 34-character canonical form with hyphens in a case of an EIDR content ID system, and the cid element is 11-character or 12-character form or ASCII characters representing a decimal value of a

4-byte unsigned integer Compact Ad-ID Identifier in a case of an Ad-ID Content ID system,; and decoding the cid element of the recovery data table.