



(51) International Patent Classification:

G06F 11/08 (2006.01) H04N 21/236 (2011.01)  
H04L 29/06 (2006.01) H04N 21/4402 (2011.01)

(21) International Application Number:

PCT/FI2015/050885

(22) International Filing Date:

15 December 2015 (15.12.2015)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/094,764 19 December 2014 (19.12.2014) US

(71) Applicant: NOKIA TECHNOLOGIES OY [FI/FI];  
Karaportti 3, 02610 Espoo (FI).

(72) Inventors: MALAMAL VADAKITAL, Vinod Kumar;  
Insinöörinkatu 54 B 38, 33720 Tampere (FI). HANNUK-  
SELA, Miska; Rusthollinrinne 2, 33610 Tampere (FI).

(74) Agents: NOKIA TECHNOLOGIES OY et al.; Ari  
Aarnio, IPR Department, Karakaari 7, 02610 Espoo (FI).

(81) Designated States (unless otherwise indicated, for every  
kind of national protection available): AE, AG, AL, AM,  
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,  
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,  
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,  
KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG,  
MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM,  
PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC,  
SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,  
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every  
kind of regional protection available): ARIPO (BW, GH,  
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,  
TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,  
TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,  
DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,  
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,  
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,  
GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the  
claims and to be republished in the event of receipt of  
amendments (Rule 48.2(h))

(54) Title: MEDIA ENCAPSULATING AND DECAPSULATING

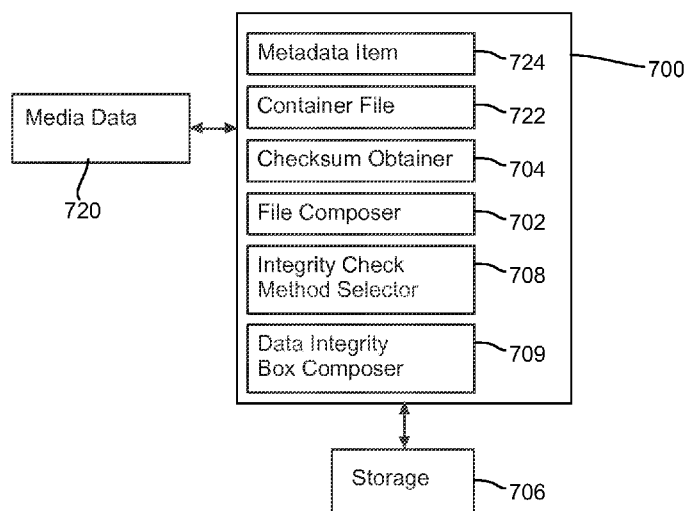


Fig. 13a

(57) Abstract: There are disclosed various methods, apparatuses and computer program products for media encapsulating and decapsulating. A container file according to a first format is received. The container file includes at least one meta data unit, and includes or refers to at least one media data unit, and the container file further includes or refers to a first checksum. Information of a first scope for the first checksum is obtained, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit. A first reproduced checksum is derived from the container file and/or the at least one media data unit on the basis of the first scope. The validity of the data within the first scope is determined on the basis of the first checksum and the first reproduced checksum; and subject to the data within the first scope being valid, the data within the first scope is processed.

## MEDIA ENCAPSULATING AND DECAPSULATING

### TECHNOLOGICAL FIELD

**[0001]** The present invention relates generally to the use of multimedia file formats. More particularly, the present invention relates to a method for encapsulating media data signal into a file and a method for decapsulating media data from a file. The present invention also relates to apparatuses and computer program products for encapsulating media data into a file and apparatuses and computer program products for decapsulating media data from a file.

### BACKGROUND

**[0002]** This section is intended to provide a background or context to the invention that is recited in the claims. The description herein may include concepts that could be pursued, but are not necessarily ones that have been previously conceived or pursued. Therefore, unless otherwise indicated herein, what is described in this section is not prior art to the description and claims in this application and is not admitted to be prior art by inclusion in this section.

**[0003]** A multimedia container file format is an element in the chain of multimedia content production, manipulation, transmission and consumption. In this context, the coding format (i.e., the elementary stream format) relates to the action of a specific coding algorithm that codes the content information into a bitstream. The container file format comprises mechanisms for organizing the generated bitstream in such a way that it can be accessed for local decoding and playback, transferring as a file, or streaming, all utilizing a variety of storage and transport architectures. The container file format can also facilitate the interchanging and editing of the media, as well as the recording of received real-time streams to a file. As such, there may be substantial differences between the coding format and the container file format.

### BRIEF SUMMARY

**[0004]** Various embodiments provide systems and methods for detecting errors or losses from multimedia files, locating errors and losses within multimedia files, and possibly making use of identified correct parts of multimedia files.

**[0005]** Various aspects of examples of the invention are provided in the detailed description.

**[0006]** According to a first aspect there is provided a method comprising:

receiving a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

obtaining information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

deriving a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

determining the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the first scope being valid, processing the data within the first scope.

**[0007]** According to a second aspect there is provided an apparatus comprising at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

receive a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

obtain information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

derive a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

determine the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the first scope being valid, process the data within the first scope.

**[0008]** According to a third aspect there is provided a computer program product embodied on a non-transitory computer readable medium, comprising computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

receive a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

obtain information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

derive a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

determine the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the first scope being valid, process the data within the first scope.

**[0009]** According to a fourth aspect there is provided a method comprising:

forming a container file according to a first format;

including the container file at least one meta data unit,

including the container file or referring in the container file to at least one media data unit and a first checksum; and

including the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.

**[0010]** According to a fifth aspect there is provided an apparatus comprising at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

form a container file according to a first format;

include the container file at least one meta data unit,

include the container file or referring in the container file to at least one media data unit and a first checksum; and

include the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.

**[0011]** According to a sixth aspect there is provided a computer program product embodied on a non-transitory computer readable medium, comprising computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

form a container file according to a first format;

include the container file at least one meta data unit,

include the container file or referring in the container file to at least one media data unit and a first checksum; and

include the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.

**[0012]** According to a seventh aspect there is provided an apparatus configured to perform the method of the first aspect.

[0013] According to an eighth aspect there is provided an apparatus configured to perform the method of the fourth aspect.

[0014] According to a ninth aspect there is provided an apparatus comprising.

means for receiving a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

means for obtaining information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

means for deriving a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

means for determining the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

means for processing the data within the first scope, subject to the data within the first scope being valid.

[0015] According to a tenth aspect there is provided an apparatus comprising:

means for forming a container file according to a first format;

means for including the container file at least one meta data unit,

means for including the container file or referring in the container file to at least one media data unit and a first checksum; and

means for including the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.

#### BRIEF DESCRIPTION OF THE DRAWING(S)

[0016] Having thus described some embodiments of the invention in general terms, reference will now be made to the accompanying drawings, which are not necessarily drawn to scale, and wherein:

[0017] Figure 1 shows an example containment hierarchy of ISOBMFF box structures;

[0018] Figure 2 illustrates a simplified file structure according to the ISO base media file format;

[0019] Figure 3 illustrates some structures of an International Organization for Standardization (ISO) base media file format;

[0020] Figure 4 illustrates an example of containment of hierarchy of a segment as a tree diagram, in accordance with an embodiment;

[0021] Figure 5 illustrates an example of a segment;

- [0022] Figure 6 illustrates the example of segment of Figure 5 with some k bytes lost in transmissions;
- [0023] Figure 7 illustrates some examples of scopes of data integrity boxes, in accordance with an embodiment;
- [0024] Figure 8 illustrates as a tree diagram an example of a containment of hierarchy of a segment transmitted over a loss prone channels, in accordance with an embodiment;
- [0025] Figure 9 illustrates an example of transmission order of boxes of Figure 8s, in accordance with an embodiment;
- [0026] Figure 10 illustrates an example of loss of boxes;
- [0027] Figure 11 illustrates another example of loss of boxes;
- [0028] Figure 12 illustrates an example of a use of a data integrity box, in accordance with an embodiment;
- [0029] Figure 13a depicts an example of an apparatus suitable for composing media files;
- [0030] Figure 13b depicts an example of an apparatus suitable for decomposing container files;
- [0031] Figure 13c depicts a flow diagram of a method for composing media files, in accordance with an embodiment;
- [0032] Figure 13d depicts a flow diagram of a method for decomposing media files, in accordance with an embodiment;
- [0033] Figure 14 depicts an example illustration of some functional blocks, formats, and interfaces included in an HTTP streaming system;
- [0034] Figure 15a illustrates an example of a regular web server operating as a HTTP streaming server;
- [0035] Figure 15b illustrates an example of a regular web server connected with a dynamic streaming server;
- [0036] Figure 16 shows schematically an electronic device employing some embodiments of the invention;
- [0037] Figure 17 shows schematically a user equipment suitable for employing some embodiments of the invention;
- [0038] Figure 18 further shows schematically electronic devices employing embodiments of the invention connected using wireless and/or wired network connections; and
- [0039] Figure 19 is a graphical representation of an example of a generic multimedia communication system within which various embodiments may be implemented.

## DETAILED DESCRIPTION

**[0040]** Some embodiments of the present invention will now be described more fully hereinafter with reference to the accompanying drawings, in which some, but not all embodiments of the invention are shown. Indeed, various embodiments of the invention may be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will satisfy applicable legal requirements. Like reference numerals refer to like elements throughout. As used herein, the terms “data,” “content,” “information” and similar terms may be used interchangeably to refer to data capable of being transmitted, received and/or stored in accordance with some embodiments of the present invention. Thus, use of any such terms should not be taken to limit the spirit and scope of embodiments of the present invention.

**[0041]** Additionally, as used herein, the term ‘circuitry’ refers to (a) hardware-only circuit implementations (e.g., implementations in analog circuitry and/or digital circuitry); (b) combinations of circuits and computer program product(s) comprising software and/or firmware instructions stored on one or more computer readable memories that work together to cause an apparatus to perform one or more functions described herein; and (c) circuits, such as, for example, a microprocessor(s) or a portion of a microprocessor(s), that require software or firmware for operation even if the software or firmware is not physically present. This definition of ‘circuitry’ applies to all uses of this term herein, including in any claims. As a further example, as used herein, the term ‘circuitry’ also includes an implementation comprising one or more processors and/or portion(s) thereof and accompanying software and/or firmware. As another example, the term ‘circuitry’ as used herein also includes, for example, a baseband integrated circuit or applications processor integrated circuit for a mobile phone or a similar integrated circuit in a server, a cellular network device, other network device, and/or other computing device.

**[0042]** As defined herein a “computer-readable storage medium,” which refers to a non-transitory, physical storage medium (e.g., volatile or non-volatile memory device), can be differentiated from a “computer-readable transmission medium,” which refers to an electromagnetic signal.

**[0043]** Some further definitions used in this specification may be described as follows. A coding format (which may also be referred to as media compression format) may relate to an action of a specific coding algorithm that codes content information into a bitstream. Media data may be defined as a sequence of bits that forms a coded representation of multimedia, e.g., coded representation of pictures, coded representation of an image or images, coded representation of audio or speech, coded representation of graphics. Coded representation of multimedia usually conforms to a coding format, such as H.264/AVC, H.265/HEVC, AMR,

etc. A container file format may comprise means of organizing the generated bitstream in such a way that it may be accessed for decoding and playback, transferred as a file, or streamed, all possibly utilizing a variety of storage and transport architectures. Furthermore, a container file format can facilitate interchange and editing of the media as well as recording of received real-time streams to a file. Metadata may be understood to comprise structural or descriptive information about media data.

**[0044]** The H.264/AVC standard was developed by the Joint Video Team (JVT) of the Video Coding Experts Group (VCEG) of the Telecommunications Standardization Sector of International Telecommunication Union (ITU-T) and the Moving Picture Experts Group (MPEG) of International Organisation for Standardization (ISO) / International Electrotechnical Commission (IEC). The H.264/AVC standard is published by both parent standardization organizations, and it is referred to as ITU-T Recommendation H.264 and ISO/IEC International Standard 14496-10, also known as MPEG-4 Part 10 Advanced Video Coding (AVC). There have been multiple versions of the H.264/AVC standard, integrating new extensions or features to the specification. These extensions include Scalable Video Coding (SVC) and Multiview Video Coding (MVC).

**[0045]** Version 1 of the High Efficiency Video Coding (H.265/HEVC a.k.a. HEVC) standard was developed by the Joint Collaborative Team – Video Coding (JCT-VC) of VCEG and MPEG. The standard was published by both parent standardization organizations, and it is referred to as ITU-T Recommendation H.265 and ISO/IEC International Standard 23008-2, also known as MPEG-H Part 2 High Efficiency Video Coding (HEVC). Version 2 of H.265/HEVC included scalable, multiview, and fidelity range extensions, which may be abbreviated SHVC, MV-HEVC, and REXT, respectively. Version 2 of H.265/HEVC was pre-published as ITU-T Recommendation H.265 (10/2014) and is likely to be published as Edition 2 of ISO/IEC 23008-2 in 2015. There are currently ongoing standardization projects to develop further extensions to H.265/HEVC, including three-dimensional and screen content coding extensions, which may be abbreviated 3D-HEVC and SCC, respectively.

**[0046]** When media data is stored using a container file format, at least part of the metadata may be represented by the file format structures of the container file format. However, a part of the metadata may be represented using a metadata format that is distinct from the container file format. For example, the metadata format and the container file format may be specified in different specifications and/or they may use different basic units or elements. For example, the container file format may be based on elements comprising key-length-value triplets, where the key indicates the type of information, the length indicates the size of the information, and the value comprises the information itself. The box structure used in the ISO Base Media File Format may be regarded as an example of a container file element

comprising of a key-length-value triplet. Continuing the same example, a metadata format may be based on an XML (Extensible Markup Language) schema.

**[0047]** Some media file format standards include ISO base media file format (ISO/IEC 14496-12, which may be abbreviated ISOBMFF), MPEG-4 file format (ISO/IEC 14496-14, also known as the MP4 format), advanced video coding (AVC) file format (ISO/IEC 14496-15), the Image File Format (ISO/IEC 23008-12) and 3GPP (3<sup>rd</sup> Generation Partnership Project) file format (3GPP TS 26.244, also known as the 3GP format). The scalable video coding (SVC) and the multiview video coding (MVC) file formats are specified as amendments to the AVC file format. The ISO file format is the base for derivation of all the above mentioned file formats, excluding the ISO file format itself. These file formats, including the ISO file format itself, may generally be called the ISO family of file formats.

**[0048]** One building block in the ISO base media file format is called a box. Each box may have a header and a payload. The box header indicates the type of the box and the size of the box in terms of bytes. A box may enclose other boxes, and the ISO file format specifies which box types are allowed within a box of a certain type. Furthermore, the presence of some boxes may be mandatory in each file, while the presence of other boxes may be optional. Additionally, for some box types, it may be allowable to have more than one box present in a file. Thus, the ISO base media file format may be considered to specify a hierarchical structure of boxes. Each box of the ISO base media file may be identified by a four character code (4CC). The header may provide information about the type and size of the box. An example containment hierarchy of ISOBMFF box structures is shown in Figure 1.

**[0049]** The ISOBMFF is designed considering a variety of use cases; such uses cases may include (a) content creation (b) interchange (c) communication and (d) local and streamed presentation. The Dynamic Adaptive Streaming over HTTP (DASH) is an example of a communication format that uses ISOBMFF for its objectives. While DASH was originally meant as a format that is transported over HTTP (Hypertext Transfer Protocol), it has also found uses in 3GPP based protocols such as MBMS (multimedia broadcast/multicast service) and DVB (Digital Video Broadcasting) based IP (Internet Protocol) broadcasts. While some of these transmission protocols are bi-directional, some are uni-directional. Furthermore, some of these transmission protocols operate in a loss prone environment.

**[0050]** According to the ISO family of file formats, a file may include media data and metadata that may be enclosed in separate boxes. In an example embodiment, the media data may be provided in a media data (mdat) box and the movie (moov) box may be used to enclose the metadata. In some cases, for a file to be operable, both of the mdat and moov boxes must be present. The movie (moov) box may include one or more tracks, and each track may reside in one corresponding track box. A track may be, for example, one of the following

types: media, hint, timed metadata. A media track refers to samples (which may also be referred to as media samples) formatted according to a media compression format (and its encapsulation to the ISO base media file format). A hint track refers to hint samples, containing cookbook instructions for constructing packets for transmission over an indicated communication protocol. The cookbook instructions may include guidance for packet header construction and may include packet payload construction. In the packet payload construction, data residing in other tracks or items may be referenced. As such, for example, data residing in other tracks or items may be indicated by a reference as to which piece of data in a particular track or item is instructed to be copied into a packet during the packet construction process. A timed metadata track may refer to samples describing referred media and/or hint samples. For the presentation of one media type, one media track may be selected. Samples of a track may be implicitly associated with sample numbers that may be incremented e.g. by 1 in the indicated decoding order of samples. The first sample in a track may be associated with sample number 1.

**[0051]** Figure 2 illustrates an example of a simplified file structure according to the ISO base media file format. As shown in Figure 2, the file 90 may include the moov box 92 and the mdat box 94 and the moov box 92 may include tracks (trak 96 and trak 98) that correspond to video and audio, respectively.

**[0052]** The ISO base media file format does not limit a presentation to be contained in one file. As such, a presentation may be comprised within several files. As an example, one file may include the metadata for the whole presentation and may thereby include all the media data to make the presentation self-contained. Other files, if used, may not be required to be formatted to ISO base media file format, and may be used to include media data, and may also include unused media data, or other information. The ISO base media file format concerns the structure of the presentation file only. The format of the media-data files may be constrained by the ISO base media file format or its derivative formats only in that the media-data in the media files is formatted as specified in the ISO base media file format or its derivative formats.

**[0053]** The ability to refer to external files may be realized through data references. In some examples, a sample description box included in each track may provide a list of sample entries, each providing detailed information about the coding type used, and any initialization information needed for that coding. All samples of a chunk and all samples of a track fragment may use the same sample entry. A chunk may be defined as a contiguous set of samples for one track. The Data Reference (dref) box, which may also be included in each track, may define an indexed list of uniform resource locators (URLs), uniform resource names (URNs), and/or self-references to the file containing the metadata. A sample entry may

point to one index of the Data Reference box, thereby indicating the file containing the samples of the respective chunk or track fragment.

**[0054]** Movie fragments may be used when recording content to ISO files in order to avoid losing data if a recording application crashes, runs out of memory space, or some other incident occurs. Without movie fragments, data loss may occur because the file format may require that all metadata, e.g., the movie box, be written in one contiguous area of the file. Furthermore, when recording a file, there may not be sufficient amount of memory space (e.g., random access memory RAM) to buffer a movie box for the size of the storage available, and re-computing the contents of a movie box when the movie is closed may be too slow. Moreover, movie fragments may enable simultaneous recording and playback of a file using a regular ISO file parser. Furthermore, a smaller duration of initial buffering may be required for progressive downloading, e.g., simultaneous reception and playback of a file when movie fragments are used and the initial movie box is smaller compared to a file with the same media content but structured without movie fragments.

**[0055]** The movie fragment feature may enable splitting the metadata that otherwise might reside in the movie box into multiple pieces. Each piece may correspond to a certain period of time of a track. In other words, the movie fragment feature may enable interleaving file metadata and media data. Consequently, the size of the movie box may be limited and the use cases mentioned above be realized.

**[0056]** In some examples, the media samples for the movie fragments may reside in an mdat box, if they are in the same file as the moov box. For the metadata of the movie fragments, however, a moof box may be provided. The moof box may include the information for a certain duration of playback time that would previously have been in the moov box. The moov box may still represent a valid movie on its own, but in addition, it may include an mvex box indicating that movie fragments will follow in the same file. The movie fragments may extend the presentation that is associated to the moov box in time.

**[0057]** Within the movie fragment there may be a set of track fragments, including anywhere from zero to a plurality per track. The track fragments may in turn include anywhere from zero to a plurality of track runs, each of which document is a contiguous run of samples for that track. Within these structures, many fields are optional and can be defaulted. The metadata that may be included in the moof box may be limited to a subset of the metadata that may be included in a moov box and may be coded differently in some cases. Details regarding the boxes that can be included in a moof box may be found from the ISO base media file format specification. A self-contained movie fragment may be defined to consist of a moof box and an mdat box that are consecutive in the file order and where the mdat box contains the samples of the movie fragment (for which the moof box provides the

metadata) and does not contain samples of any other movie fragment (i.e. any other moov box).

**[0058]** The ISO Base Media File Format contains three mechanisms for timed metadata that can be associated with particular samples: sample groups, timed metadata tracks, and sample auxiliary information. Derived specification may provide similar functionality with one or more of these three mechanisms.

**[0059]** A sample grouping in the ISO base media file format and its derivatives, such as the AVC file format and the SVC file format, may be defined as an assignment of each sample in a track to be a member of one sample group, based on a grouping criterion. A sample group in a sample grouping is not limited to being contiguous samples and may contain non-adjacent samples. As there may be more than one sample grouping for the samples in a track, each sample grouping may have a type field to indicate the type of grouping. Sample groupings may be represented by two linked data structures: (1) a SampleToGroup box (sbgp box) represents the assignment of samples to sample groups; and (2) a SampleGroupDescription box (sgpd box) contains a sample group entry for each sample group describing the properties of the group. There may be multiple instances of the SampleToGroup and SampleGroupDescription boxes based on different grouping criteria. These may be distinguished by a type field used to indicate the type of grouping.

**[0060]** The sample group boxes (SampleGroupDescription Box and SampleToGroup Box) reside within the sample table (stbl) box, which is enclosed in the media information (minf), media (mdia), and track (trak) boxes (in that order) within a movie (moov) box. The SampleToGroup box is allowed to reside in a movie fragment. Hence, sample grouping can be done fragment by fragment.

**[0061]** Sample groups are defined in ISOBMFF using two correlated boxes. These boxes are the SampleToGroupBox, which may be indicated e.g. by a four character code 'sbgp' (fourCC: 'sbgp') and the SampleGroupDescriptionBox (fourCC: 'sgpd'). Both these boxes are located in the SampleTableBox (fourCC: 'stbl'). The SampleToGroupBox assigns a subset of samples in the track to one or more sample groups. The definition and description of each sample group defined on the track may be handled by the SampleGroupDescriptionBox.

**[0062]** Linking of samples in the track to one or more metadata schemas in the track may be handled by sample grouping. The definition of a SampleToGroupBox is defined in ISOBMFF in the following way:

```
aligned(8) class SampleToGroupBox
extends FullBox('sbgp', version, 0)
{
    unsigned int(32) grouping_type;
```

```

    if (version == 1) {
        unsigned int(32) grouping_type_parameter;
    }
    unsigned int(32) entry_count;
    for (i=1; i <= entry_count; i++) {
        unsigned int(32) sample_count;
        unsigned int(32) group_description_index;
    }
}

```

**[0063]** While the SampleToGroupBox may assign samples to sample groups, the description of the characteristics of the sample groups itself may be provided by the SampleGroupDescriptionBox. The SampleGroupDescriptionBox may be defined as follows.

```

aligned(8) class SampleGroupDescriptionBox (unsigned int(32) handler_type)
extends FullBox('sgpd', version, 0){
    unsigned int(32) grouping_type;
    if (version==1) {
        unsigned int(32) default_length;
    }
    unsigned int(32) entry_count;
    int i;
    for (i = 1 ; i <= entry_count ; i++){
        if (version==1) {
            if (default_length==0) {
                unsigned int(32) description_length;
            }
        }
    }
    switch (handler_type){
        case 'pict': // for picture tracks
            VisualSampleGroupEntry (grouping_type);
            break;
        case 'vide': // for video tracks
            VisualSampleGroupEntry (grouping_type);
            break;
        case 'soun': // for audio tracks
            AudioSampleGroupEntry(grouping_type);
            break;
        case 'hint': // for hint tracks

```

```

        HintSampleGroupEntry(grouping_type);
        break;
    }
}

```

**[0064]** The input to the SampleGroupDescriptionBox is a handler type which may be the track handler\_type. For a sequence of untimed images or image bursts the handler type may be ‘pict’ and for timed video and images the handler types may be ‘vide’. In both these cases the VisualSampleGroupEntry structure may be used. The VisualSampleGroupEntry is an extension of the abstract structure SampleGroupEntry. It is this structure where the characteristics of the sample group description entry may be placed.

**[0065]** Sample auxiliary information may be stored anywhere in the same file as the sample data itself. For self-contained media files, where the media data resides in the same file as the file format metadata (such as the moov box), sample auxiliary information may reside in a MediaData box. Sample auxiliary information may be stored (a) in multiple chunks, with the number of samples per chunk, as well as the number of chunks, matching the chunking of the primary sample data, or (b) in a single chunk for all the samples in a movie sample table (or a movie fragment). The Sample Auxiliary Information for all samples contained within a single chunk (or track run) may be stored contiguously (similarly to sample data). The Sample Auxiliary Information Sizes box may optionally include the type and a type parameter of the auxiliary information and may specify the size of the auxiliary information for each sample. The Sample Auxiliary Information Offsets box may specify the position information for the sample auxiliary information in a way similar to the chunk offsets for sample data.

**[0066]** Sample groups and timed metadata are less tightly coupled to the media data and may be ‘descriptive’, whereas sample auxiliary information might be required for decoding. Sample auxiliary information may be intended for use where the information is directly related to the sample on a one-to-one basis, and may be required for the media sample processing and presentation.

**[0067]** Figure 3 illustrates some structures of an International Organization for Standardization (ISO) base media file format (ISOBMFF). In the example of Figure 3, the metadata may be stored separately from the media data, which may be stored in one or more external files. The metadata may be partitioned into fragments covering a certain playback duration. If the file contains tracks that are alternatives to each other, such as the same content coded with different bitrate, the example of Figure 3 illustrates the case of a single metadata file for all versions. However, if the file contains tracks that are not alternatives to each other, then the example structure provides for one metadata file for each version. According to Figure 3, the media content may be stored as a metadata file and one or more media data

file(s). The metadata file may comprise an ftyp box. The ftyp box may be the first box in the metadata file and may comprise information about the brand and version number of the media content. The moov box may comprise information about the structure of the media data file. The moov box may comprise one or more track boxes, which describe the different media tracks that are described by the metadata file. The track boxes may further comprise information about the codecs and formats of the media described by the track. The track boxes may not comprise a description of the samples themselves, however, so as to keep the box relatively small in size. The track boxes may also comprise dref boxes, which may include a reference to the media data file that contains samples for the track. The metadata file may further comprise an mvex box, which may hold default values for the subsequent movie fragment boxes and may indicate that movie fragments are used. The moof box may comprise metadata that describes the samples of a movie fragment. The moof box may be structured as a plurality of traf boxes, which describe specific tracks of the movie. The traf boxes may comprise at least one trun box, which describes the media data fragments in track runs. Offsets may be provided to point to the specific media data fragment in the referenced media data file.

**[0068]** The transport file formats or segment formats that may be employed can be coarsely categorized into different classes. In one example class, transmitted files may be compliant with an existing file format that can be used for live file playback. For example, transmitted files may be compliant with the ISO base media file format or the progressive download profile of the Third Generation Partnership Project (3GPP) file format. In another example class, transmitted files may be similar to files formatted according to an existing file format used for live file playback. For example, transmitted files may be fragments of a server file, which might not be self-containing for playback individually. In another approach, files to be transmitted may be compliant with an existing file format that can be used for live file playback, but the files may be transmitted only partially and hence playback of such files may require awareness and capability of managing partial files.

**[0069]** In addition to timed metadata tracks, ISO files may contain any non-timed metadata objects in a meta box (fourCC: 'meta'). The meta box may reside at the top level of the file, within a movie box (fourCC: 'moov'), and within a track box (fourCC: 'trak'), but at most one meta box may occur at each of the file level, movie level, or track level. The meta box may be required to contain a 'hdlr' box indicating the structure or format of the 'meta' box contents. The meta box may list and characterize any number of metadata items that can be referred and each one of them can be associated with a file name and are uniquely identified with the file by item identifier (item\_id) which is an integer value. The metadata items may be for example stored in the idat box of the meta box or in an mdat box or reside in a separate file. If the metadata is located external to the file then its location may be declared by the

DataInformationBox (fourCC: 'dinf'). In the specific case that the metadata is formatted using XML syntax and is required to be stored directly in the MetaBox, the metadata may be encapsulated into either the XMLBox (fourCC: 'xml ') or the BinaryXMLBox (fourcc: 'bxml'). An item may be stored as a contiguous byte range, or it may be stored in several extents, each being a contiguous byte range. In other words, items may be stored fragmented into extents, e.g. to enable interleaving. An extent is a contiguous subset of the bytes of the resource; the resource can be formed by concatenating the extents.

**[0070]** In order to support more than one meta box at any level of the hierarchy (file, movie, or track), a meta box container box ('meco') may be used as one ISO base media file format. The meta box container box may carry any number of additional meta boxes at any level of the hierarchy (file, movie, or track). This may allow that e.g. the same meta-data is being presented in two different, alternative meta-data systems. The meta box relation box ('mere') may enable describing how different meta boxes relate to each other, e.g. whether they contain exactly the same metadata (but described with different schemes) or if one represents a superset of another one.

**[0071]** The ISO BMFF includes the so-called level mechanism to specify subsets of the file. Levels follow the dependency hierarchy so that samples mapped to level  $n$  may depend on any samples of levels  $m$ , where  $m \leq n$ , and do not depend on any samples of levels  $p$ , where  $p > n$ . For example, levels can be specified according to temporal sub-layer (e.g., temporal\_id of SVC or MVC or TemporalId of HEVC). Levels may be announced in the Level Assignment ('leva') box contained in the Movie Extends ('mvex') box. Levels cannot be specified for the initial movie. When the Level Assignment box is present, it applies to all movie fragments subsequent to the initial movie. For the context of the Level Assignment box, a fraction is defined to consist of one or more Movie Fragment boxes and the associated Media Data boxes, possibly including only an initial part of the last Media Data Box. Within a fraction, data for each level appears contiguously. Data for levels within a fraction appears in increasing order of level value. All data in a fraction shall be assigned to levels. The Level Assignment box provides a mapping from features, such as scalability layers, to levels. A feature can be specified through a track, a sub-track within a track, or a sample grouping of a track. The Level Assignment box includes the syntax element padding\_flag. padding\_flag is equal to 1 indicates that a conforming fraction can be formed by concatenating any positive integer number of levels within a fraction and padding the last Media Data box by zero bytes up to the full size that is indicated in the header of the last Media Data box. For example, padding\_flag can be set equal to 1 when each fraction contains two or more AVC, SVC, or MVC tracks of the same video bitstream, the samples for each track of a fraction are contiguous and in decoding order in a Media Data box, and the samples of the first AVC,

SVC, or MVC level contain extractor NAL units for including the video coding NAL units from the other levels of the same fraction.

**[0072]** H.264/AVC and HEVC enable encoding a bitstream in a temporally hierarchical manner, such that a temporal sub-layer can be assigned for each coded picture or access unit, where the temporal sub-layer can be for example associated with a variable TemporalId (or equivalently syntax element temporal\_id in H.264/AVC). The bitstream created by excluding all coded pictures or access units having a TemporalId greater than or equal to a selected value and including all other coded picture or access units remains conforming. Consequently, a picture or access unit having TemporalId equal to TID does not use any picture or access unit having a TemporalId greater than TID as inter prediction reference. A sub-layer or a temporal sub-layer may be defined to be a temporal scalable layer of a temporal scalable bitstream, comprising pictures or access units with a particular value of the TemporalId variable.

**[0073]** In dynamic adaptive streaming over HTTP (DASH), the multimedia content may be captured and stored on an HTTP server and may be delivered using HTTP. The content may be stored on the server in two parts: Media Presentation Description (MPD), which describes a manifest of the available content, its various alternatives, their URL addresses, and other characteristics; and segments, which contain the actual multimedia bitstreams in the form of chunks, in single or multiple files. To play the content, the DASH client may obtain the MPD e.g. by using HTTP, email, thumb drive, broadcast, or other transport methods. By parsing the MPD, the DASH client may become aware of the program timing, media-content availability, media types, resolutions, minimum and maximum bandwidths, and the existence of various encoded alternatives of multimedia components, accessibility features and required digital rights management (DRM), media-component locations on the network, and other content characteristics. Using this information, the DASH client may select the appropriate encoded alternative and start streaming the content by fetching the segments using e.g. HTTP GET requests. After appropriate buffering to allow for network throughput variations, the client may continue fetching the subsequent segments and also monitor the network bandwidth fluctuations. The client may decide how to adapt to the available bandwidth by fetching segments of different alternatives (with lower or higher bitrates) to maintain an adequate buffer.

**[0074]** The media presentation description (MPD) may provide information for clients to establish a dynamic adaptive streaming over HTTP. MPD may contain information describing media presentation, such as an HTTP- uniform resource locator (URL) of each Segment to make GET Segment request. In DASH, hierarchical data model may be used to structure media presentation as shown in Figure 14. A media presentation may comprise a sequence of

one or more Periods, each Period may contain one or more Groups, each Group may contain one or more Adaptation Sets, each Adaptation Set may contain one or more Representations, and each Representation may comprise one or more Segments. A Representation is one of the alternative choices of the media content or a subset thereof which may differ by the encoding choice, e.g. by bitrate, resolution, language, codec, etc. The Segment may contain certain duration of media data, and metadata to decode and present the included media content. A Segment may be identified by a uniform resource identifier (URI) and can be requested by a HTTP GET request. A Segment may be defined as a unit of data associated with an HTTP-URL and optionally a byte range that are specified by an MPD.

**[0075]** A DASH service may be provided as an on-demand service or live service. In the former, the MPD is a static and all Segments of a Media Presentation are already available when a content provider publishes an MPD. In the latter, however, the MPD may be static or dynamic depending on the Segment URLs construction method employed by a MPD and Segments may be created continuously as the content is produced and published to DASH clients by a content provider. Segment URLs construction method may be either template-based Segment URLs construction method or the Segment list generation method. In the former, a DASH client may be able to construct Segment URLs without updating an MPD before requesting a Segment. In the latter, a DASH client may need to periodically download the updated MPDs to get Segment URLs. For live service, hence, the template-based Segment URLs construction method may be superior to the Segment list generation method.

**[0076]** An Initialization Segment may be defined as a Segment containing metadata that is necessary to present the media streams encapsulated in Media Segments. In ISO/BMFF based segment formats, an Initialization Segment may comprise the Movie Box ('moov') which might not include metadata for any samples, i.e. any metadata for samples is provided in 'moof' boxes.

**[0077]** A Media Segment may be defined as a Segment that complies with the container file format and/or the media format or formats in use and enables playback when combined with zero or more preceding segments, and an Initialization Segment (if any). A Media Segment may contain certain duration of media data for playback at a normal speed, such duration may be referred as Media Segment duration or Segment duration. The content producer or service provider may select the Segment duration according to the desired characteristics of the service. For example, a relatively short Segment duration may be used in a live service to achieve a short end-to-end latency. The reason is that Segment duration may be a lower bound on the end-to-end latency perceived by a DASH client since a Segment is a discrete unit of generating media data for DASH. Content generation may be done in such a manner that a whole Segment of media data is made available for a server. Furthermore, many

client implementations may use a Segment as the unit for GET requests. Thus, in some arrangements for live services a Segment can be requested by a DASH client only when the whole duration of Media Segment is available as well as encoded and encapsulated into a Segment. For on-demand service, different strategies of selecting Segment duration may be used.

**[0078]** A Segment may further be partitioned into Subsegments each of which may contain complete access units. Subsegments may be indexed by Segment index, which contains information to map presentation time range and byte range for each Subsegment and may be used to make a HTTP GET request for a specific Subsegment using byte range HTTP request. If relatively long Segment duration is used, then Subsegments may be used to keep the size of HTTP responses reasonable and flexible for bitrate adaptation. A subsegment may be defined as a self-contained set of one or more consecutive movie fragments, where the self-contained set contains one or more Movie Fragment boxes with the corresponding Media Data box(es), and a Media Data Box containing data referenced by a Movie Fragment Box must follow that Movie Fragment box and precede the next Movie Fragment box containing information about the same track.

**[0079]** Each media segment may be assigned a unique URI or URL (possibly with byte range), an index, and explicit or implicit start time and duration. Each media segment may contain at least one stream access point, which is a random access or switch-to point in the media stream where decoding can start using only data from that point forward.

**[0080]** To enable downloading segments in multiple parts, a method of signaling subsegments using a segment index box may be utilized. This box describes subsegments and stream access points in the segment by signaling their durations and byte offsets. The DASH client may use the indexing information to request subsegments using partial HTTP GET requests. The indexing information of a segment may be put in the single box at the beginning of that segment, or spread among many indexing boxes in the segment. Different methods of spreading are possible, such as hierarchical, daisy chain, and hybrid. This technique may avoid adding a large box at the beginning of the segment and therefore may prevent a possible initial download delay. MPEG-DASH defines segment-container formats for both ISO Base Media File Format and MPEG-2 Transport Streams.

**[0081]** The Subsegment Index box ('ssix') provides a mapping from levels (as specified by the Level Assignment box) to byte ranges of the indexed subsegment. In other words, this box provides a compact index for how the data in a subsegment is ordered according to levels into partial subsegments. It enables a client to easily access data for partial subsegments by downloading ranges of data in the subsegment. When the Subsegment Index box is present, each byte in the subsegment is assigned to a level. If the range is not associated with any

information in the level assignment, then any level that is not included in the level assignment may be used. There is 0 or 1 Subsegment Index boxes present per each Segment Index box that indexes only leaf subsegments, i.e. that only indexes subsegments but no segment indexes. A Subsegment Index box, if any, is the next box after the associated Segment Index box. A Subsegment Index box documents the subsegment that is indicated in the immediately preceding Segment Index box. Each level may be assigned to exactly one partial subsegment, i.e. byte ranges for one level are contiguous. Levels of partial subsegments are assigned by increasing numbers within a subsegment, i.e., samples of a partial subsegment may depend on any samples of preceding partial subsegments in the same subsegment, but not the other way around. For example, each partial subsegment contains samples having an identical temporal sub-layer and partial subsegments appear in increasing temporal sub-layer order within the subsegment. When a partial subsegment is accessed in this way, the final Media Data box may be incomplete, that is, less data is accessed than the length indication of the Media Data Box indicates is present. The length of the Media Data box may need adjusting, or padding may be used. The padding\_flag in the Level Assignment Box indicates whether this missing data can be replaced by zeros. If not, the sample data for samples assigned to levels that are not accessed is not present, and care should be taken not to attempt to process such samples.

**[0082]** DASH specifies different timelines including Media Presentation timeline and Segment availability times. The former indicates the presentation time of an access unit with a media content which is mapped to the global common presentation timeline. Media Presentation timeline may enable DASH to seamlessly synchronize different media components which are encoded with different coding techniques and share a common timeline. The latter indicates a wall-clock time and is used to signal clients the availability time of Segments which may be identified by HTTP URLs. A DASH client may be able to identify an availability time of a certain Segment by comparing the wall-clock time to the Segment availability time assigned to that Segment. Segment availability time may be used in live delivery of media Segments, referred as live service. For live service, the Segment availability time is different from Segment to Segment and a certain Segment's availability time may depend on the position of the Segment in the Media Presentation timeline. For on-demand service, the Segment availability time may be the same for all Segments.

**[0083]** DASH supports rate adaptation by dynamically requesting Media Segments and/or Subsegments from different Representations within an Adaptation Set to match varying network bandwidth. When a DASH client switches up/down Representation, coding dependencies within Representation may need to be taken into account. In media decoding, a Representation switch may only happen at a random access point (RAP), which may be used in video coding techniques such as H.264/AVC. In order to avoid requesting and transmitting of media data that will not be decoded, RAPs may be aligned at the beginning of Media

Segments and/or Subsegments, and the MPD and/or the segment index box may be used to indicate alignment of RAPs at the beginning of Media Segments and/or Subsegments. Consequently, DASH clients may be able to conclude which Segments and/or Subsegments to request so that when Representation switching is performed the first Segment and/or Subsegment of a destination Representation starts with a RAP and the Segments and/or Subsegments of the source and destination Representation are aligned (time-wise). In DASH, a more general concept named Stream Access Point (SAP) is introduced to provide a codec-independent solution for accessing a Representation and switching between Representations. In DASH, a SAP is specified as a position in a Representation that enables playback of a media stream to be started using only the information contained in Representation data starting from that position onwards (preceded by initialising data in the Initialisation Segment, if any). Hence, Representation switching can be performed in SAP.

**[0084]** A content provider may create Segment and Subsegment of multiple Representations in a way that may make switching simpler. In a simple case, each Segment and Subsegment starts with a SAP and the boundaries of Segment and Subsegment are aligned across the Representation of an Adaptation Set. In such a case a DASH client may be able to switch Representations without error drift by requesting Segments or Subsegments from an original Representation to a new Representation. In DASH, restrictions to construct Segment and Subsegment are specified in MPD and Segment Index in order to facilitate a DASH client to switch Representations without introducing an error drift. One of the usages of profile specified in DASH is to provide different levels of restrictions to construct Segment and Subsegment etc.

**[0085]** The Matroska file format is capable of (but not limited to) storing any of video, audio, picture, or subtitle tracks in one file. Matroska file extensions include .mkv for video (with subtitles and audio), .mk3d for stereoscopic video, .mka for audio-only files, and .mks for subtitles only. Matroska may be used as a basis format for derived file formats, such as WebM.

**[0086]** Matroska uses Extensible Binary Meta Language (EBML) as basis. EBML specifies a binary and octet (byte) aligned format inspired by the principle of XML. EBML itself is a generalized description of the technique of binary markup. A Matroska file consists of Elements that make up an EBML "document." Elements incorporate an Element ID, a descriptor for the size of the element, and the binary data itself. Elements can be nested.

**[0087]** A Segment Element of Matroska is a container for other top-level (level 1) elements. A Matroska file may comprise (but is not limited to be composed of) one Segment. Multimedia data in Matroska files is organized in Clusters (or Cluster Elements), each containing typically a few seconds of multimedia data. A Cluster comprises BlockGroup

elements, which in turn comprise Block Elements. A Cues Element comprises metadata which may assist in random access or seeking and may include file pointers or respective timestamps for seek points.

**[0088]** When Matroska files are carried as DASH segments or alike, the association of DASH units and Matroska units may be specified as follows. A subsegment (of DASH) may be defined as one or more consecutive Clusters of Matroska-encapsulated content. An Initialization Segment of DASH may be required to comprise the EBML header, Segment header (of Matroska), Segment Information (of Matroska) and Tracks, and may optionally comprise other level1 elements and padding. A Segment Index of DASH may comprise a Cues Element of Matroska.

**[0089]** MPEG Media Transport (MMT), specified in ISO/IEC 23008-1, comprises protocol and format specifications in three functional areas: encapsulation, signaling, and delivery. In the encapsulation function, encoded media, e.g. video bitstreams, is encapsulated into a specified MMT format (or respectively decapsulated from the MMT format). The MMT format defines the logical structure of the content and the physical encapsulation format by using the ISOBMFF. The signaling function defines message formats to manage MMT package consumption and delivery. The delivery function defines an application-layer protocol, MMTP, that supports packetized streaming delivery of the media encapsulated into MMT format, delivery of any files (e.g. similarly to file delivery over the FLUTE protocol), and delivery of MMT signaling information. It is noted that media encapsulated into MMT format may additionally or alternatively be transmitted over other protocols than MMTP, such as HTTP.

**[0090]** A Media Processing Unit (MPU) in MMT may be defined as a media data item that may be processed by an MMT entity and consumed by the presentation engine independently from other MPUs. For the streaming delivery, MMT defines a new brand 'mpuf' of the ISOBMFF allowing only media track in the file, enforcing the use of movie fragments, and requiring the use of the 'tfdt' box to indicate earliest decode time of every fragment. In the streaming delivery, an MPU may be an ISOBMFF conformant file with the 'mpuf' brand included in the 'ftyp' box.

**[0091]** The delivery of MPUs to MMT receivers using the MMT protocol involves a packetization and depacketization procedure to take place at the MMT sending entity and MMT receiving entity, respectively. The packetization procedure transforms an MPU into a set of MMTP payloads that are then carried in MMTP packets. The MMTP payload format allows for fragmentation of the MMTP payload to enable the delivery of large payloads. It also allows for the aggregation of multiple MMTP payload data units into a single MMTP payload, to cater for smaller data units. At the receiving entity depacketization is performed to

recover the original MPU data. Several depacketization modes are defined to address the different requirements of the overlaying applications.

**[0092]** MMTP includes a payload format for carrying an ISOBMFF file (using the 'mpuf' brand), which targets at enabling the carriage of the file over an unreliable connection such as UDP/IP and suits multicast delivery. MMTP enables indicating the nature and priority of the fragment that is carried. A fragment of an MPU may either be MPU metadata, a Movie Fragment metadata, an MFU, or a non-timed media data item. The MPU metadata comprises the ftyp, mmpu, and moov boxes and may comprise other file-level boxes, such as the meta box. The movie fragment metadata comprises the moof box and the header of the respective mdat box, excluding all media data inside the mdat box. The MFU may be considered to comprise a sample or sub-sample of timed media data or an item of non-timed media data. MMTP enables out-of-order delivery of media samples (MFUs) before their metadata (i.e., Movie Fragment metadata).

**[0093]** A hash function may be defined as is any function that can be used to map digital data of arbitrary size to digital data of fixed size, with slight differences in input data possibly producing big differences in output data. A cryptographic hash function may be defined as a hash function that is intended to be practically impossible to invert, i.e. to create the input data based on the hash value alone. Cryptographic hash function may comprise e.g. the MD5 function. An MD5 value may be a null-terminated string of UTF-8 characters containing a base64 encoded MD5 digest of the input data. One method of calculating the string is specified in IETF RFC 1864. It should be understood that instead of or in addition to MD5, other types of integrity check schemes could be used in various embodiments, such as different forms of the cyclic redundancy check (CRC), such as the CRC scheme used in ITU-T Recommendation H.271.

**[0094]** A checksum or hash sum may be defined as a small-size datum from an arbitrary block of digital data which may be used for the purpose of detecting errors which may have been introduced during its transmission or storage. The actual procedure which yields the checksum, given a data input may be called a checksum function or checksum algorithm. A checksum algorithm will usually output a significantly different value, even for small changes made to the input. This is especially true of cryptographic hash functions, which may be used to detect many data corruption errors and verify overall data integrity; if the computed checksum for the current data input matches the stored value of a previously computed checksum, there is a high probability the data has not been altered or corrupted. Check digits and parity bits are special cases of checksums, appropriate for small blocks of data. Some error-correcting codes are based on special checksums which not only detect common errors but also allow the original data to be recovered in certain cases.

**[0095]** The term checksum may be defined to be equivalent to a cryptographic hash value or alike.

**[0096]** A file segment may be defined as a contiguous range of bytes within a file. A file segment may but need not relate to Segments (as defined in DASH), Subsegments (as defined in DASH), file structures, such as self-contained movie fragments, or alike. For example, a file segment may (but needs not to) correspond to the byte range of a self-contained movie fragment or the byte range of a Subsegment.

**[0097]** Data integrity refers to assuring the accuracy of data, i.e., whether the data has not been altered or corrupted. The overall intent of any data integrity technique is to ensure the data is the same as it was when it was originally recorded.

**[0098]** Multimedia files may be erroneous or incomplete due to multiple reasons, including but not limited to the following. Multimedia files may be transmitted using a protocol stack that does not guarantee error-free delivery. For example, DASH segments may be carried over the MBMS file delivery protocol stack, which despite the included forward error correction (FEC) means does not guarantee error-free reception.

**[0099]** Protocol stack implementations may cause loss of data. For example, while HTTP over TCP may include an automated retransmission request (ARQ) mechanism for loss recovery, the size of the reception buffer may be limited and hence buffer overflows may happen and/or an attempt to process a certain piece of received media data may happen prior to its reception.

**[0100]** Interfaces and/or formats used to convey multimedia files may lack means to indicate losses, errors, or incomplete reception of a multimedia file. For example, a DASH client may access ISOBMFF-formatted segments e.g. from the MBMS protocol stack and pass the obtained ISOBMFF-formatted segments to a media player. However, the interface to the media player as well as the format of the segments may lack means to indicate losses, errors, or incomplete reception.

**[0101]** Multimedia file formats may have many optional parts (e.g. optional boxes in ISOBMFF) and may be extensible (e.g. new boxes can be introduced in the ISOBMFF). Sometimes a proper interpretation or use of an optional part may require the presence of another optional part of the file. A file editing unit might not be knowledgeable of all the optional features of a file format. Hence, a file editing unit may keep an optional part (e.g. an optional box of ISOBMFF) in the file, while the contents of the optional part may depend on something that have been edited by the file editing unit and hence the contents of the optional part may no longer be valid.

**[0102]** Mass storage, such as hard drives, solid state drives and optical discs, may suffer from data corruption. Such data corruption may become an issue particularly with large video

files. It would be desirable to detect individual corrupted coded pictures prior to attempting to decode them. Also it may be desirable not to discard a whole file, if only some coded pictures are corrupted.

**[0103]** A problem related to losses or non-received data in multimedia files may be that media data can be misaligned when compared to the metadata using byte offsets or such to refer to the media data or other pieces of metadata.

**[0104]** Communication of media formatted using ISOBMFF may be done using many different protocols and many of them operate in loss prone transmission environments. While the transmission protocols themselves may provide error recovery mechanisms such as ARQ and FEC, there is no guarantee that such error correction method is always successful. When uncorrected errors are encountered in an ISOBMFF based media transmission, they can render large parts of the transmitted media unusable. While the errors may not corrupt the actual media data but only some purely informative part information about the media data, the ISOBMFF's sample locating mechanism may reference the wrong data thus losing the correctly received media data.

**[0105]** One problem that an unrecoverable data loss can cause in the consumption of ISOBMFF formatted media can be illustrated using the following example. For this example, one segment is considered. The containment of hierarchy of a segment may be represented using the tree diagram shown in Figure 4.

**[0106]** It may further be considered that a segment as illustrated by Figure 5 is transmitted over a loss prone network. The sample data contained in the 'mdat' of this segment is referenced using data offset fields in the 'tfhd' and 'trun' box. It is also assumed that for a sample contained in this segment the computed data offset is some integer value  $x$  computed from the first byte of the 'moof' box.

**[0107]** Upon transmission of the segment over a loss prone network it may be assumed that some part of the 'moof' is un-recoverably lost. This is shown in the illustration. For example, it is now assumed that the amount of data lost is some  $k$  bytes. However, the receiver does not know how many bytes are actually lost. This is illustrated in Figure 6.

**[0108]** In this scenario all of the sample data is actually correctly received. However, due to the loss of data ( $k$  bytes) in the 'moof', the sample is now actually located at offset  $(x-k)$  and not at  $x$  as is signaled by the fields in the 'tfhd' and 'trun' boxes. This may lead to the incorrect sample referencing and the receiver might have no option but to discard the samples of the entire segment.

**[0109]** In accordance with an embodiment, selective integrity protection information may be used to enable determining if required parts of a file or a file segment, such as a self-contained movie fragment, a segment (as defined in DASH), or a subsegment (as defined in

DASH), have been received and/or are correct in content. In a file creator, a first cryptographic hash value (a.k.a. a first checksum), such as MD5 or CRC value, is derived from a subset of the data contained in and/or referred to by a container structure of the file, and the first cryptographic hash value is included in the container structure along with an indication of the data over which the cryptographic hash is computed, the indication hereinafter referred to as the hash source indication or the first scope for the first checksum. In a file reader, an indication of the data over which the cryptographic hash is computed is parsed from the file, i.e. information of the first scope of the first checksum is obtained from the file. A first reproduced cryptographic hash value is computed over the received data identified by the first scope, the first cryptographic hash value is parsed from the file, and the first cryptographic hash value and the first reproduced cryptographic hash value are compared. The data over which the cryptographic hash is computed is a subset of the data contained in and/or referred to by the container structure. In an embodiment, the first scope is one or more entities of the file format, such as boxes and/or samples of ISOBMFF. The first cryptographic hash value together with the hash source indication may facilitate protection of the meta data and media data units of the file that may be required for processing (e.g. decoding or playback) of the file or a subset of the file.

**[0110]** In an embodiment, as a response to the first cryptographic hash value being equal to the first reproduced cryptographic hash value, the data within the first scope is determined to be valid or correct.

**[0111]** In an embodiment, as a response to the determining that the data within the first scope is valid, the data within the first scope is processed. Processing may e.g. involve parsing and/or decoding of the data.

**[0112]** In an embodiment, as a response to the first cryptographic hash value being equal to the first reproduced cryptographic hash value, the data within the first scope is determined to be invalid or incorrect.

**[0113]** In an embodiment, as a response to the determining that the data within the first scope is invalid, the processing of the data within the first scope is omitted. Omitting may for example include continuing the processing from data following the first scope.

**[0114]** In an embodiment, which may be applied in addition to or instead of other embodiments, as a response to the determining that the data within the first scope is invalid, the data within the first scope is overwritten with a default value, such as zero. The default value may cause a file reader to omit or gracefully process the data within the first scope even if the file reader did not derive the first reproduced cryptographic hash value or determine whether the data within the first scope is valid (through comparing the first cryptographic hash value and the first reproduced cryptographic hash value).

**[0115]** A scope for a checksum function may be defined to refer to parts of the container file that is input to a checksum function or checksum algorithm and yields a single value of the checksum. Checksum values can be generated for different parts of the container file, even overlapping parts. Parts of the container file that produces a checksum need not be sequentially ordered in the file. In other words, a scope for a checksum function need not be a contiguous range of bytes within the container file. For example, the scope may comprise a subset of media samples that need not be consequent in decoding order, where the subset of media samples may be indicated for example using the sample grouping mechanism. For example, a video bitstream may be temporally scalable, i.e. contain more than one temporal sub-layer, pictures of one temporal sub-layer may be interleaved with pictures of another temporal sub-layer in decoding or file order, and the scope may be defined to correspond to the pictures of one temporal sub-layer. For example, the temporal level ('tele') sample grouping may be used to indicate the mapping of pictures to temporal sub-layers, and the scope may be indicated with reference to one or more sample group description indices of the 'tele' sample grouping, indicating which temporal sub-layers are covered by the scope. In another example, the container file comprises more than one track, such as a video track and an audio track, whose samples are interleaved in the 'mdat' box. In this case, the scope may comprise the samples of only one track and hence might not be sequentially ordered in the file.

**[0116]** In an embodiment, the first scope may be one or more of the following:

- An initialization segment (as defined in/for DASH) or alike.
- The movie box ('moov') or alike.
- The movie box ('moov') and the 'mdat' containing the media data of the movie box.
- A positive integer number of complete segments (as defined in/for DASH). E.g. one segment.
- A moof box ('moof').
- A positive integer number of self-contained movie fragments. E.g. one self-contained movie fragment.
- A positive integer number of complete subsegments (as defined in/for DASH), which may be required to be contained in the same segment. E.g. one subsegment.
- A positive integer number of levels (as defined in the context of the Level Assignment box or alike) or partial subsegments (as defined in the context of the Subsegment Index box or alike) or alike.
- A positive integer number of samples (e.g. within an 'mdat' box) or alike. The samples may be consecutive e.g. in the file order or decoding order, or may be specifically indicated, e.g. using a sample grouping mechanism.
- A positive integer number of sub-samples (e.g. within an 'mdat' box) or alike.

- A positive integer number of items (as defined in ISOBMFF) or alike.
- A positive integer number of item extents (as defined in ISOBMFF) or alike.

**[0117]** A fragment of an MPU (as defined in MMT) or alike, which may e.g. be MPU metadata, a Movie Fragment metadata, an MFU, or a non-timed media data item.

**[0118]** In an embodiment, a file writer can select the data over which the cryptographic hash is computed in more than one way and selects the data over which the cryptographic hash is computed. Additionally, the file writer indicates, in one or more syntax elements in the file, the first scope.

**[0119]** In an embodiment, a composition file is generated by concatenating received files or file segments. For example, fragment of MPUs may be concatenated. Additionally, other embodiments are applied to the composition file to determinate validity of data as indicated by the hash source indication.

**[0120]** In accordance with an exemplary technical implementation, a syntax structure, such as a box, that provides the length and sanity check (a.k.a. integrity check) of each of the encapsulated syntax structures, such as boxes, in the order that they were transmitted or included in the file is introduced. A receiver may use this box intelligently to decipher the location and the amount of bytes lost or corrupted, if any, and to use concepts such as padding so that the media sample data references are not misaligned.

**[0121]** In accordance with an embodiment, the elementary unit for the sanity check may be a container box called data integrity box (DataIntegrityBox ('dint')). It may contain within it one and only one integrity header box (IntegrityHeaderBox ('inth')) and zero or more integrity check boxes (IntegrityCheck ('ichk')). The integrity header box may establish all the common information required for decoding the contents of this box. The integrity check box is for signaling a cryptographic hash along with a switchable mechanism to indicate the data over which the cryptographic hash is computed.

**[0122]** In the following, the description of the data integrity box is provided, in accordance with an embodiment. The box type is DataIntegrityBox ('dint') and it may be located at the root level or in any box that contains other boxes in the ISOBMFF or its derived formats. The data integrity box is not mandatory, wherein a file may contain zero or more data integrity boxes.

**[0123]** An example of a syntax of the data integrity box may be as follows:

```
aligned(8) class DataIntegrityBox()
extends Box('dint') {
}
```

**[0124]** The integrity header box 'inth' signals all the information common to the boxes encapsulated in the data integrity box 'dint'. Such information may include the algorithm used

for computing the cryptographic hash, and the switch\_code used by the integrity header boxes in the data integrity box. Three flavors of the data integrity box, discriminated using the flags input field of the data integrity box, are envisioned.

**[0125]** When a data integrity box contains an integrity header box with flags field set to zero, it may indicate to the reader that the switch\_code in the integrity header box is 'boxt', and the integrity is computed for only those boxes at the same hierarchical containment level as the data integrity box itself.

**[0126]** There may be a maximum of only one data integrity box per box that contains other boxes with an integrity header box with flags field set to zero.

**[0127]** When a data integrity box contains an integrity header box with flags field set to one, it may indicate to the reader that the switch\_code in the data integrity box is 'boxt', and the integrity is computed for boxes that are one level lower in hierarchy than the data integrity box.

**[0128]** There can be multiple data integrity boxes with integrity header flags field set to one. Figure 7 shows the scope of the data integrity boxes 900 when it contains data integrity boxes with flag field set to zero or one, in accordance with an embodiment.

**[0129]** When a data integrity box contains an integrity header box with flags field set to two, it may indicate to the reader that the switch\_code in the integrity header box is one of {'stsz', 'trak', 'sgpd'}, and the hash is computed over specific types of sample information, as will be described below in connection with the description of the integrity check box.

**[0130]** There may be only one data integrity box with integrity header flags field set to two and switch\_field set to 'stsz' or 'trak'. There may be multiple data integrity boxes with integrity header flags field set to two and switch\_field set to 'sgpd'. This type of data integrity box may be used to provide the sanity check information of coded samples carried in the file.

**[0131]** In the following, the description of the integrity header box is provided, in accordance with an embodiment. The box name is IntegrityHeaderBox (integrity header) and it may be located in the data integrity box (DataIntegrityBox). The number of integrity header boxes in the data integrity box is one.

**[0132]** An example of a syntax of the integrity header box may be as follows:

```
aligned(8) class IntegrityHeaderBox ()
extends FullBox(integrity header, version = 0, flags) {
    unsigned int(32) algorithm;
    unsigned int(32) switch_code;
    if (flags == 1) {
        unsigned int(32) parent_fourcc
    }
```

}

**[0133]** The semantics of the fields of the integrity header box may be as follows, in accordance with an embodiment.

**[0134]** The valid values of flags field may be zero, one, or two. Other values may be reserved. When the switch\_code field is set to 'boxt', the flags discriminate whether the boxes over which the cryptographic hash is computed is at the same containment level or a level below. When flags field is set to one, the boxes accounted for is one level below in hierarchy, while flags field set to zero signals that the boxes accounted for are at the same level of hierarchy as this box. The flags field value of two indicates that switch\_code is one of {'stsz', 'trak', 'sgpd'}. A data integrity box with the integrity header box with flags field set to two can be contained only in the 'stbl' or 'traf' box.

**[0135]** The algorithm field signals the algorithm used for computing the cryptographic hash. The algorithm is specified using fourCC codes. For example, when MD5 is used, the fourCC code that the algorithm field is set to is 'md5'.

**[0136]** The value of the switch\_code field is a fourCC code that signals the input data over which the cryptographic hash is computed and communicated in the data integrity check boxes of this data integrity box. The following fourCC codes may be used:

**[0137]** The first code is 'boxt', which indicates that the hash string is computed over the concatenated bytes of a box which is signalled in the integrity check boxes in the data integrity box.

**[0138]** The second code is 'stsz', which indicates that the hash string is computed over the concatenated 32-bit sizes of the samples in the track

**[0139]** The third code is 'trak', which indicates that the hash string is computed over the concatenated bytes of samples that are in the track.

**[0140]** The fourth code is 'sgpd', which indicates that the hash string is computed over concatenated bytes of samples that are mapped to any group of the indicated type.

**[0141]** When the flags field is set to one, the boxes used for computing the hash is one level lower in hierarchy. The value of the parent\_4cc field is the fourCC of the box, in the same level of hierarchy as the data integrity box, whose child boxes are used for computing the hash.

**[0142]** The integrity check boxes 'ichk' may signal the computed hash for data signalled with a particular switch\_code. The data over which the hash is computed may depend on the switch\_code communicated in the integrity header box. When the integrity header box signals a 'boxt' switch\_code, this box informs the box over which the hash is computed, the size of the box over which the hash is computed, and the hash itself. A data integrity box with an

integrity check box that uses any other switch other than 'boxt' will be placed in the 'stbl' box or the 'traf' box. This data integrity box will have its integrity header flag field set to two.

**[0143]** In the following, the description of the integrity check box is provided, in accordance with an embodiment. The box type is IntegrityCheck ('ichk') and it may be located in the data integrity box (DataIntegrityBox). The integrity check box is not mandatory, wherein a file may contain zero or more integrity check boxes.

**[0144]** An example of a syntax of the integrity check box may be as follows:

```
aligned(8) class IntegrityCheck ()
extends FullBox ('ichk', version = 0, 0) {
    string hash_string;
    if (switch_code == 'boxt') {
        unsigned int(32) box_4cc;
        unsigned int(64) box_size;
    }
    if (switch_code == 'sgpd') {
        unsigned int(32) grouping_type;
        unsigned int(32) num_entries;
        for(i=0; i<num_entries; i++) {
            unsigned int(32) group_description_index[i];
        }
    }
}
```

**[0145]** The semantics of the fields of the integrity check box may be as follows, in accordance with an embodiment.

**[0146]** The box\_4cc field contains the fourCC of the box for which the cryptographic hash is computed.

**[0147]** The box\_size field contains the length of the box over which the cryptographic hash is computed.

**[0148]** If sample groups are used for the hash calculation, the grouping\_type field reflects the grouping\_type of the sample groups considered for the hash string calculations.

**[0149]** When the num\_entries field is equal to 0, it specifies that the cryptographic hash is derived from all samples mapped to the identified sample group. When the num\_entries field is greater than 0, it specifies that the cryptographic hash is derived from samples mapped to indicated sample group description indices.

**[0150]** When the `group_description_index[i]` is present, it specifies that the MD5 checksum is derived from samples mapped to the sample group description index equal to `group_description_index[i]`.

**[0151]** In the following, some information is summarized regarding the use of the data integrity box, the integrity header box and the integrity check box. There shall be only one data integrity box having an integrity header box with the flags field set to zero in any box that contains other boxes. There may be more than one data integrity box having an integrity header box with flags field set to one or two.

**[0152]** A data integrity box having an integrity header box with a flags field set to zero shall be the first box in order of containment, apart from any other boxes that has already been mandated to occur earlier.

**[0153]** A data integrity box, regardless of the value of the flags field, accounts for all boxes in the hierarchy over which the hash is computed; this includes any data integrity boxes in that particular hierarchy.

**[0154]** The coding of the list of integrity check boxes in the data integrity box is in the same exact order as they occur in the contained box.

**[0155]** A data integrity box with integrity check box that uses any other switch than 'boxt' may be in the 'stbl' box or the 'traf' box and this data integrity box may have its flags field set to two.

**[0156]** In the following, some exemplary usages of the data integrity box are provided in more detail.

**[0157]** Example 1:

**[0158]** For this example, an ISOBMFF segment is considered having a box hierarchy as shown in Figure 8. The data integrity boxes with its integrity header box flag fields set to zero are inserted at each hierarchy. As the rules for the usage of the data integrity boxes state, data integrity boxes with an integrity header box having its flags field set to zero shall record the hash for only those boxes at the same level. Only one data integrity box with an integrity header box having its flags field set to zero is allowed per hierarchical level. The data integrity box contains integrity check boxes that compute the hash for all boxes in the same hierarchical level and in the same order of their appearance. Figure 9 shows the hierarchical level of the boxes and the order of transmission.

**[0159]** The arrows 900 in the Figure 9 show the scope of the data integrity box at that level. For example, the data integrity box 901 at level 1 has in its scope the ordered sequences of boxes ['styp', 'dint', 'sidx', 'moof', 'mdat'], the data integrity box 902 at level 2 has in its scope the ordered sequences of boxes ['mfhd', 'dint', 'traf'], and the data integrity box 903 at level 3 has in its scope the ordered sequences of boxes ['tfhd', 'dint', 'traf', 'sbgd', 'sgpd'],

‘subs’]. Next, it is assumed that some part of the segment shown in Figure 9 is lost in transmission. The loss of the data in terms of the boxes lost is illustrated in Figure 10. In this example, the ‘sbgp’ and the ‘sgpd’ boxes are lost. They account for ‘k’ bytes of data. Such a loss would make a sample that was coded with a sample offset of x bytes to be now at (x-k) bytes. However, since a receiver does not know how many bytes are lost in transmission it is not able to correctly synchronize to the sample data.

**[0160]** Without the use of data integrity box, the receiver has two options to deal with such a loss. The first is to drop the entire segment because it is not able to synchronize to the samples correctly. The other is to parse the loss effect segment byte stream for the fourCC codes of boxes to and to try to fix the data references.

**[0161]** With the availability of the data integrity box the exact location may be found with a high probability. In the loss example shown in Figure 10, a receiver would get the entire segment and compute the hash for each of the boxes received at level 1. It should find that the computed hash of the received ‘moof’ box and the one that is present in the data integrity box 901 at the level 1 is not the same. Hence, it knows that the error is in the ‘moof’ box. Next it computes, for each of the boxes in the ‘moof’ box, the hash of the received data and compares it with what has been recorded in the data integrity box 902 in the ‘moof’ box. It will find that the hash of the ‘traf’ code does not match with the value stored, while the others match. So it ascertains that the error is in the ‘traf’ box. Proceeding similarly into level 3, it can identify on the basis of difference between the calculated hash value and the hash value indicated in the data integrity box 903 at level 3 that it is the ‘sbgp’ and the ‘sgpd’ boxes that are actually lost. If the receiver finds that the information in these lost boxes are not really necessary for its application the receiver can pad these lost boxes with dummy data with the length of bytes as given by the dint box in the ‘traf’ box. This will make any data references computed in the ‘tfhd’ and the ‘trun’ boxes to be correct and decoding can be correctly done.

**[0162]** Example 2:

**[0163]** Next the same segment as in the Example 1 is considered, except that the order of boxes in at level 3 is different. The order of the boxes in this example is as illustrated in Figure 11. The loss scenario has also changed in this example. Here, it is assumed that even the data integrity box 903 at level 3 has been lost. In such a case it may be difficult for a receiver to clearly demarcate the lost boxes at that level. This is when the data integrity box with flags set to one becomes useful. If this same segment was also sent along with data integrity boxes with flags set to one as shown in Figure 12, the data integrity boxes 902, 903 at the lower level are redundantly coded at the next higher level, as is illustrated with data integrity boxes 904 and 905 at level 1 and level 2, respectively. This redundancy may be used to correctly position the lost boxes of the segment.

**[0164]** In accordance with an exemplary technical implementation, the subsegment index box or alike is appended to include checksums for each partial subsegment. For example, the following syntax may be used:

```
aligned(8) class SubsegmentIndexBox extends FullBox('ssix', 0, 0) {
    unsigned int(32) subsegment_count;
    for( i=1; i <= subsegment_count; i++)
    {
        unsigned int(32) range_count;
        for ( j=1; j <= range_count; j++) {
            unsigned int(8) level;
            unsigned int(24) range_size;

        }
    }
    // the syntax above is identical to that specified in ISOBMFF
    for( i=1; i <= subsegment_count; i++)
    {
        unsigned int(8) level_count;
        for ( j=1; j <= level_count; j++) {
            unsigned int(8) md5_level;
            string md5_value;

        }
    }
}
```

**[0165]** The semantics for the syntax above may be specified for example as follows: subsegment\_count is a positive integer specifying the number of subsegments for which partial subsegment information is specified in this box. subsegment\_count may be required to be equal to reference\_count (i.e., the number of movie fragment references) in the immediately preceding Segment Index box. range\_count specifies the number of partial subsegment levels into which the media data is grouped. range\_size indicates the size of the partial subsegment. level specifies the level to which this partial subsegment is assigned. level\_count specifies the number of levels for which the MD5 checksum information is provided. md5\_level specifies the value of the level for which the following MD5 checksum (md5\_value) is provided. md5\_value is a null-terminated string of UTF-8 characters containing a base64 encoded MD5 digest of the input data. The method of calculating the string is specified in IETF RFC 1864. The input data consists of the partial subsegment(s) associated with this level value through the level and range\_size syntax elements.

**[0166]** In the following some usage of the data integrity box in various transmission scenarios will be described in more detail.

**[0167]** Many transmission protocols/formats may use ISOBMFF and its derived specification for transmission of media data. Some of them communicate over full duplex channels (e.g. DASH over HTTP) and some of them communicate over uni-directional channels (e.g. DASH over MBMS). The data integrity box and its component boxes could be used in any such channels tweaking the way they are communicated. In uni-directional communication protocols, the data integrity box could be inserted in-line with the actual

media transmission so that a receiver could use the integrity check data to identify losses as and when they occur.

**[0168]** In bi-directional channels the data integrity box and its component boxes could be used in two ways. A first method is that a transmitting device transmits them in-line with the media data so that a receiving device, upon detecting a loss, can identify the location of the loss and request only the lost bytes and not the entire segment. In a second method, the data integrity box and its component boxes can be stored at a server end and the receiving device may request the data integrity boxes, which may be packaged by a transmitting device only when losses are detected. This may avoid the necessity to retransmit large amounts of media data which may have been correctly received.

**[0169]** In the following, some example embodiments are described in more detail with reference to the apparatuses of Figures 13a and 13b and flow diagrams of Figures 13c and 13d. According to an example embodiment, the apparatus 700 may receive or otherwise obtain media data 720 for generating one or more container files from the media data (block 750 in Figure 13c). The media data may comprise one or more media data units. Each media data unit may relate to video, audio, image or other media data (e.g. text). The apparatus 700 may have a file composer 702 or other means for obtaining and/or generating a container file 722. The file composer 702 may include one or more media data units in the container file 722, or the file composer 702 may include a reference to the one or more media data units in the container file 722. The reference may, for example, comprise a URN (Uniform Resource Name) or URL (Uniform Resource Locator) that identifies a file or a resource containing the one or more media data units. Said file or resource may reside in the same or different storage system, mass memory and/or apparatus than the container file 722.

**[0170]** The file composer 702 of the apparatus 700 or some other means may further instruct an integrity check method selector 708 to determine 752 whether one or more data integrity boxes 901—905 shall be included in the container file 722 and if so, at which level or levels of the hierarchy of the file system the data integrity box(es) will be placed and which kind of scope of the data they will cover (block 754). For example, the integrity check method selector 708 may decide to insert one data integrity box at the highest level (i.e. level 1 in the example hierarchy of ISOBMFF) and to use data on the same level of the hierarchy, i.e. the scope of the calculation of the integrity data is within the same level. Hence, a data integrity box composer 709 may use selected boxes on level 1 to calculate 756 a checksum (e.g. a hash) and compose 758 the integrity header box for the data integrity box accordingly. In this example, the flags field is set to zero in the integrity header box, the switch\_code is 'boxt', and the integrity header box is further included with the checksum which may be calculated e.g. by the checksum obtainer 704 and information on the scope i.e. which boxes have been used

in the checksum calculation. Accordingly, if the data integrity box composer 709 selects to use boxes on a lower level than the level in which the data integrity box will be located, the checksum obtainer 704 may calculate a checksum on the basis of one or more boxes at the lower level, wherein the data integrity box composer 709 may set the flags field to one in the integrity header box, the switch\_code is 'boxt', and include the checksum and information on the scope to the integrity header box. Furthermore, another option for the data integrity box composer 709 may be to decide to use specific types of sample information in the checksum calculation, wherein the data integrity box composer 709 may set the flags field to two, and the switch\_code in the integrity header box may be one of {'stsz', 'trak', 'sgpd'}, It should be noted that the above described options are examples only and it may be possible to use other kinds of indications in the integrity header box.

**[0171]** The calculated or otherwise obtained checksum and the scope information may be included 760, for example, in the container file 722. Alternatively or additionally, the calculated or otherwise obtained checksum and the scope information may be included, for example, in a file or resource containing the metadata item. Alternatively or additionally, the calculated or otherwise obtained checksum and the scope information may be included, for example, in a file or resource referred to by the container file, for example using a URN or a URL.

**[0172]** The container file 722 and other information may be stored into the memory 706 and/or to some other storage location, and/or the container file 722 may be transmitted to another device e.g. for storing, decomposing and playback.

**[0173]** In accordance with an embodiment, a file reader 710 (Figure 13b) or player may perform the following operations to decompose the media data from the container file 722 and to check the validity of the media data. The file reader 710 may comprise an integrity check method verifier 718, which may examine (block 770 in Figure 13d) whether the container file 722 contains or refers to any data integrity boxes. If the integrity check method verifier 718 detects that the container file 722 comprises or refers to at least one data integrity box, the integrity check method verifier 718 examines 772 the integrity header box of the data integrity box, which is the scope of the checksum contained in the data integrity box. As was described earlier in this specification, the value of the flags field and, depending on value of the flags field, the value of the switch\_code may define the scope. The integrity check method verifier 718 may indicate the scope to a checksum verifier 714 or other means for deriving 774 a checksum from the media units in the file as indicated in the scope information. The checksum verifier 714 may compare 778 the derived checksum to the checksum stored in the container file 722. If the two checksums are equal, the metadata item may be determined to be valid 780. Otherwise, the metadata item may be determined to be not valid 782. In accordance

with an embodiment, instead of or in addition to parsing the scope information and/or the stored checksum from the container file, the scope information and/or the stored checksum may be parsed from the file containing the metadata item and/or from a file or resource referred to by the container file, for example using a URN or a URL.

**[0174]** If the checksum verifier 714 detects that the meta data items are not valid, the file decomposer 712 may examine 784 whether non-valid parts (e.g. missing parts) may be ignored or not. If the file decomposer 712 determines that the container file 722 may still be decomposed and utilize the correctly received parts, the file decomposer 712 may decompose 786 those validly received parts. If, on the other hand, the file decomposer 712 determines that the container file 722 may not be validly decomposed, the file decomposer 712 may, for example, request 788 a transmitting device (e.g. the apparatus 700) to retransmit the container file 722 or parts of it.

**[0175]** When an apparatus (700, 710, or alike) accesses files or resources other than the container file e.g. using a URI or URN, and the files or resources do not reside in the same apparatus as the container file, the accessing may be performed for example using a network connection, such as a wired, wireless (e.g. WLAN) and/or mobile (e.g. GSM, 3G, and/or LTE/4G) connection.

**[0176]** In Figure 14 an example illustration of some functional blocks, formats, and interfaces included in a hypertext transfer protocol (HTTP) streaming system are shown. A file encapsulator 100 takes media bitstreams of a media presentation as input. The bitstreams may already be encapsulated in one or more container files 102. The bitstreams may be received by the file encapsulator 100 while they are being created by one or more media encoders. The file encapsulator converts the media bitstreams into one or more files 104, which can be processed by a streaming server 110 such as the HTTP streaming server. The output 106 of the file encapsulator is formatted according to a server file format. The HTTP streaming server 110 may receive requests from a streaming client 120 such as the HTTP streaming client. The requests may be included in a message or messages according to e.g. the hypertext transfer protocol such as a GET request message. The request may include an address indicative of the requested media stream. The address may be the so called uniform resource locator (URL). The HTTP streaming server 110 may respond to the request by transmitting the requested media file(s) and other information such as the metadata file(s) to the HTTP streaming client 120. The HTTP streaming client 120 may then convert the media file(s) to a file format suitable for play back by the HTTP streaming client and/or by a media player 130. The converted media data file(s) may also be stored into a memory 140 and/or to another kind of storage medium. The HTTP streaming client and/or the media player may

include or be operationally connected to one or more media decoders, which may decode the bitstreams contained in the HTTP responses into a format that can be rendered.

**[0177]** Server File Format

**[0178]** A server file format is used for files that the HTTP streaming server 110 manages and uses to create responses for HTTP requests. There may be, for example, the following three approaches for storing media data into file(s).

**[0179]** In a first approach a single metadata file is created for all versions. The metadata of all versions (e.g. for different bitrates) of the content (media data) resides in the same file. The media data may be partitioned into fragments covering certain playback ranges of the presentation. The media data can reside in the same file or can be located in one or more external files referred to by the metadata.

**[0180]** In a second approach one metadata file is created for each version. The metadata of a single version of the content resides in the same file. The media data may be partitioned into fragments covering certain playback ranges of the presentation. The media data can reside in the same file or can be located in one or more external files referred to by the metadata.

**[0181]** In a third approach one file is created per each fragment. The metadata and respective media data of each fragment covering a certain playback range of a presentation and each version of the content resides in their own files. Such chunking of the content to a large set of small files may be used in a possible realization of static HTTP streaming. For example, chunking of a content file of duration 20 minutes and with 10 possible representations (5 different video bitrates and 2 different audio languages) into small content pieces of 1 second, would result in 12000 small files. This may constitute a burden on web servers, which has to deal with such a large amount of small files.

**[0182]** The first and the second approach i.e. a single metadata file for all versions and one metadata file for each version, respectively, are illustrated in Figure 3 using the structures of the ISO base media file format. In the example of Figure 3, the metadata is stored separately from the media data, which is stored in external file(s). The metadata is partitioned into fragments 207a, 214a; 207b, 214b covering a certain playback duration. If the file contains tracks 207a, 207b that are alternatives to each other, such as the same content coded with different bitrates, Figure 3 illustrates the case of a single metadata file for all versions; otherwise, it illustrates the case of one metadata file for each version.

**[0183]** HTTP Streaming Server

**[0184]** A HTTP streaming server 110 takes one or more files of a media presentation as input. The input files are formatted according to a server file format. The HTTP streaming server 110 responds 114 to HTTP requests 112 from a HTTP streaming client 120 by encapsulating media in HTTP responses. The HTTP streaming server outputs and transmits a

file or many files of the media presentation formatted according to a transport file format and encapsulated in HTTP responses.

**[0185]** In accordance with an embodiment, the HTTP streaming servers 110 can be coarsely categorized into three classes. The first class is a web server, which is also known as a HTTP server, in a “static” mode. In this mode, the HTTP streaming client 120 may request one or more of the files of the presentation, which may be formatted according to the server file format, to be transmitted entirely or partly. The server is not required to prepare the content by any means. Instead, the content preparation is done in advance, possibly offline, by a separate entity. Figure 15a illustrates an example of a web server as a HTTP streaming server. A content provider 300 may provide a content for content preparation 310 and an announcement of the content to a service/content announcement service 320. The user device 330, which may contain the HTTP streaming client 120, may receive information regarding the announcements from the service/content announcement service 320 wherein the user of the user device 330 may select a content for reception. The service/content announcement service 320 may provide a web interface and consequently the user device 330 may select a content for reception through a web browser in the user device 330. Alternatively or in addition, the service/content announcement service 320 may use other means and protocols such as the Service Advertising Protocol (SAP), the Really Simple Syndication (RSS) protocol, or an Electronic Service Guide (ESG) mechanism of a broadcast television system. The user device 330 may contain a service/content discovery element 332 to receive information relating to services/contents and e.g. provide the information to a display of the user device. The streaming client 120 may then communicate with the web server 340 to inform the web server 340 of the content the user has selected for downloading. The web server 340 may then fetch the content from the content preparation service 310 and provide the content to the HTTP streaming client 120.

**[0186]** The second class is a (regular) web server operationally connected with a dynamic streaming server as illustrated in Figure 15b. The dynamic streaming server 410 dynamically tailors the streamed content to a client 420 based on requests from the client 420. The HTTP streaming server 430 interprets the HTTP GET request from the client 420 and identifies the requested media samples from a given content. The HTTP streaming server 430 then locates the requested media samples in the content file(s) or from the live stream. It then extracts and envelopes the requested media samples in a container 440. Subsequently, the newly formed container with the media samples is delivered to the client in the HTTP GET response body.

**[0187]** The first interface “1” in Figures 15a and 15b is based on the HTTP protocol and defines the syntax and semantics of the HTTP Streaming requests and responses. The HTTP Streaming requests/responses may be based on the HTTP GET requests/responses.

**[0188]** The second interface “2” in Figure 15b enables access to the content delivery description. The content delivery description, which may also be called as a media presentation description, may be provided by the content provider 450 or the service provider. It gives information about the means to access the related content. In particular, it describes if the content is accessible via HTTP Streaming and how to perform the access. The content delivery description is usually retrieved via HTTP GET requests/responses but may be conveyed by other means too, such as by using SAP, RSS, or ESG.

**[0189]** The third interface “3” in Figure 15b represents the Common Gateway Interface (CGI), which is a standardized and widely deployed interface between web servers and dynamic content creation servers. Other interfaces such as a representational State Transfer (REST) interface are possible and would enable the construction of more cache-friendly resource locators.

**[0190]** The Common Gateway Interface (CGI) defines how web server software can delegate the generation of web pages to a console application. Such applications are known as CGI scripts; they can be written in any programming language, although scripting languages are often used. One task of a web server is to respond to requests for web pages issued by clients (usually web browsers) by analyzing the content of the request, determining an appropriate document to send in response, and providing the document to the client. If the request identifies a file on disk, the server can return the contents of the file. Alternatively, the content of the document can be composed on the fly. One way of doing this is to let a console application compute the document's contents, and inform the web server to use that console application. CGI specifies which information is communicated between the web server and such a console application, and how.

**[0191]** The representational State Transfer is a style of software architecture for distributed hypermedia systems such as the World Wide Web (WWW). REST-style architectures consist of clients and servers. Clients initiate requests to servers; servers process requests and return appropriate responses. Requests and responses are built around the transfer of "representations" of "resources". A resource can be essentially any coherent and meaningful concept that may be addressed. A representation of a resource may be a document that captures the current or intended state of a resource. At any particular time, a client can either be transitioning between application states or at rest. A client in a rest state is able to interact with its user, but creates no load and consumes no per-client storage on the set of servers or on the network. The client may begin to send requests when it is ready to transition to a new state. While one or more requests are outstanding, the client is considered to be transitioning states. The representation of each application state contains links that may be used next time the client chooses to initiate a new state transition.

**[0192]** The third class of the HTTP streaming servers according to this example classification is a dynamic HTTP streaming server. Otherwise similar to the second class, but the HTTP server and the dynamic streaming server form a single component. In addition, a dynamic HTTP streaming server may be state-keeping.

**[0193]** Server-end solutions can realize HTTP streaming in two modes of operation: static HTTP streaming and dynamic HTTP streaming. In the static HTTP streaming case, the content is prepared in advance or independent of the server. The structure of the media data is not modified by the server to suit the clients' needs. A regular web server in "static" mode can only operate in static HTTP streaming mode. In the dynamic HTTP streaming case, the content preparation is done dynamically at the server upon receiving a non-cached request. A regular web server operationally connected with a dynamic streaming server and a dynamic HTTP streaming server can be operated in the dynamic HTTP streaming mode.

**[0194]** Transport File Format

**[0195]** In an example embodiment transport file formats can be coarsely categorized into two classes. In the first class transmitted files are compliant with an existing file format that can be used for file playback. For example, transmitted files are compliant with the ISO Base Media File Format or the progressive download profile of the 3GPP file format.

**[0196]** In the second class transmitted files are similar to files formatted according to an existing file format used for file playback. For example, transmitted files may be fragments of a server file, which might not be self-containing for playback individually. In another approach, files to be transmitted are compliant with an existing file format that can be used for file playback, but the files are transmitted only partially and hence playback of such files requires awareness and capability of managing partial files.

**[0197]** Transmitted files can usually be converted to comply with an existing file format used for file playback.

**[0198]** HTTP Cache

**[0199]** An HTTP cache 150 (Figure 14) may be a regular web cache that stores HTTP requests and responses to the requests to reduce bandwidth usage, server load, and perceived lag. If an HTTP cache contains a particular HTTP request and its response, it may serve the requestor instead of the HTTP streaming server.

**[0200]** HTTP streaming client

**[0201]** An HTTP streaming client 120 receives the file(s) of the media presentation. The HTTP streaming client 120 may contain or may be operationally connected to a media player 130 which parses the files, decodes the included media streams and renders the decoded media streams. The media player 130 may also store the received file(s) for further use. An interchange file format can be used for storage.

**[0202]** In some example embodiments the HTTP streaming clients can be coarsely categorized into at least the following two classes. In the first class conventional progressive downloading clients guess or conclude a suitable buffering time for the digital media files being received and start the media rendering after this buffering time. Conventional progressive downloading clients do not create requests related to bitrate adaptation of the media presentation.

**[0203]** In the second class active HTTP streaming clients monitor the buffering status of the presentation in the HTTP streaming client and may create requests related to bitrate adaptation in order to guarantee rendering of the presentation without interruptions.

**[0204]** The HTTP streaming client 120 may convert the received HTTP response payloads formatted according to the transport file format to one or more files formatted according to an interchange file format. The conversion may happen as the HTTP responses are received, i.e. an HTTP response is written to a media file as soon as it has been received. Alternatively, the conversion may happen when multiple HTTP responses up to all HTTP responses for a streaming session have been received.

**[0205]** Interchange file formats

**[0206]** In some example embodiments the interchange file formats can be coarsely categorized into at least the following two classes. In the first class the received files are stored as such according to the transport file format.

**[0207]** In the second class the received files are stored according to an existing file format used for file playback.

**[0208]** A media file player

**[0209]** A media file player 130 may parse, verify, decode, and render stored files. A media file player 130 may be capable of parsing, verifying, decoding, and rendering either or both classes of interchange files. A media file player 130 is referred to as a legacy player if it can parse and play files stored according to an existing file format but might not play files stored according to the transport file format. A media file player 130 is referred to as an HTTP streaming aware player if it can parse and play files stored according to the transport file format.

**[0210]** In some implementations, an HTTP streaming client merely receives and stores one or more files but does not play them. In contrast, a media file player parses, verifies, decodes, and renders these files while they are being received and stored.

**[0211]** In some implementations, the HTTP streaming client 120 and the media file player 130 are or reside in different devices. In some implementations, the HTTP streaming client 120 transmits a media file formatted according to a interchange file format over a network connection, such as a wireless local area network (WLAN) connection, to the media file

player 130, which plays the media file. The media file may be transmitted while it is being created in the process of converting the received HTTP responses to the media file.

Alternatively, the media file may be transmitted after it has been completed in the process of converting the received HTTP responses to the media file. The media file player 130 may decode and play the media file while it is being received. For example, the media file player 130 may download the media file progressively using an HTTP GET request from the HTTP streaming client. Alternatively, the media file player 130 may decode and play the media file after it has been completely received.

**[0212]** HTTP pipelining is a technique in which multiple HTTP requests are written out to a single socket without waiting for the corresponding responses. Since it may be possible to fit several HTTP requests in the same transmission packet such as a transmission control protocol (TCP) packet, HTTP pipelining allows fewer transmission packets to be sent over the network, which may reduce the network load.

**[0213]** A connection may be identified by a quadruplet of server IP address, server port number, client IP address, and client port number. Multiple simultaneous TCP connections from the same client to the same server are possible since each client process is assigned a different port number. Thus, even if all TCP connections access the same server process (such as the Web server process at port 80 dedicated for HTTP), they all have a different client socket and represent unique connections. This is what enables several simultaneous requests to the same Web site from the same computer.

**[0214]** Categorization of Multimedia Formats

**[0215]** The multimedia container file format is an element used in the chain of multimedia content production, manipulation, transmission and consumption. There may be substantial differences between a coding format (also known as an elementary stream format) and a container file format. The coding format relates to the action of a specific coding algorithm that codes the content information into a bitstream. The container file format comprises means of organizing the generated bitstream in such way that it can be accessed for local decoding and playback, transferred as a file, or streamed, all utilizing a variety of storage and transport architectures. Furthermore, the file format can facilitate interchange and editing of the media as well as recording of received real-time streams to a file. An example of the hierarchy of multimedia file formats is described in Figure 1.

**[0216]** Figure 16 is a graphical representation of an example of a generic multimedia communication system within which various embodiments may be implemented. As shown in Figure 16, a data source 1500 provides a source signal in an analog, uncompressed digital, or compressed digital format, or any combination of these formats. An encoder 1510 encodes the source signal into a coded media bitstream. It should be noted that a bitstream to be decoded

can be received directly or indirectly from a remote device located within virtually any type of network. Additionally, the bitstream can be received from local hardware or software. The encoder 1510 may be capable of encoding more than one media type, such as audio and video, or more than one encoder 1510 may be required to code different media types of the source signal. The encoder 1510 may also get synthetically produced input, such as graphics and text, or it may be capable of producing coded bitstreams of synthetic media. In the following, only processing of one coded media bitstream of one media type is considered to simplify the description. It should be noted, however, that typically real-time broadcast services comprise several streams (typically at least one audio, video and text sub-titling stream). It should also be noted that the system may include many encoders, but in Figure 16 only one encoder 1510 is represented to simplify the description without a lack of generality. It should be further understood that, although text and examples contained herein may specifically describe an encoding process, one skilled in the art would understand that the same concepts and principles also apply to the corresponding decoding process and vice versa.

**[0217]** The coded media bitstream is transferred to a storage 1520. The storage 1520 may comprise any type of mass memory to store the coded media bitstream. The format of the coded media bitstream in the storage 1520 may be an elementary self-contained bitstream format, or one or more coded media bitstreams may be encapsulated into a container file. Some systems operate “live”, i.e. omit storage and transfer coded media bitstream from the encoder 1510 directly to the sender 1530. The coded media bitstream is then transferred to the sender 1530, also referred to as the server, on a need basis. The format used in the transmission may be an elementary self-contained bitstream format, a packet stream format, or one or more coded media bitstreams may be encapsulated into a container file. The encoder 1510, the storage 1520, and the sender 1530 may reside in the same physical device or they may be included in separate devices. The encoder 1510 and sender 1530 may operate with live real-time content, in which case the coded media bitstream is typically not stored permanently, but rather buffered for small periods of time in the content encoder 1510 and/or in the sender 1530 to smooth out variations in processing delay, transfer delay, and coded media bitrate.

**[0218]** The sender 1530 sends the coded media bitstream using a communication protocol stack. The stack may include but is not limited to Real-Time Transport Protocol (RTP), User Datagram Protocol (UDP), and Internet Protocol (IP). When the communication protocol stack is packet-oriented, the sender 1530 encapsulates the coded media bitstream into packets. For example, when RTP is used, the sender 1530 encapsulates the coded media bitstream into RTP packets according to an RTP payload format. Typically, each media type has a dedicated

RTP payload format. It should be again noted that a system may contain more than one sender 1530, but for the sake of simplicity, the following description only considers one sender 1530.

**[0219]** If the media content is encapsulated in a container file for the storage 1520 or for inputting the data to the sender 1530, the sender 1530 may comprise or be operationally attached to a "sending file parser" (not shown in the figure). In particular, if the container file is not transmitted as such but at least one of the contained coded media bitstream is encapsulated for transport over a communication protocol, a sending file parser locates appropriate parts of the coded media bitstream to be conveyed over the communication protocol. The sending file parser may also help in creating the correct format for the communication protocol, such as packet headers and payloads. The multimedia container file may contain encapsulation instructions, such as hint tracks in the ISO Base Media File Format, for encapsulation of the at least one of the contained media bitstream on the communication protocol.

**[0220]** The sender 1530 may or may not be connected to a gateway 1540 through a communication network. The gateway 1540 may perform different types of functions, such as translation of a packet stream according to one communication protocol stack to another communication protocol stack, merging and forking of data streams, and manipulation of data stream according to the downlink and/or receiver capabilities, such as controlling the bit rate of the forwarded stream according to prevailing downlink network conditions. Examples of gateways 1540 include MCUs, gateways between circuit-switched and packet-switched video telephony, Push-to-talk over Cellular (PoC) servers, IP encapsulators in digital video broadcasting-handheld (DVB-H) systems, or set-top boxes that forward broadcast transmissions locally to home wireless networks. When RTP is used, the gateway 1540 is called an RTP mixer or an RTP translator and typically acts as an endpoint of an RTP connection.

**[0221]** The system includes one or more receivers 1550, typically capable of receiving, demodulating, and de-capsulating the transmitted signal into a coded media bitstream. The coded media bitstream is transferred to a recording storage 1555. The recording storage 1555 may comprise any type of mass memory to store the coded media bitstream. The recording storage 1555 may alternatively or additively comprise computation memory, such as random access memory. The format of the coded media bitstream in the recording storage 1555 may be an elementary self-contained bitstream format, or one or more coded media bitstreams may be encapsulated into a container file. If there are multiple coded media bitstreams, such as an audio stream and a video stream, associated with each other, a container file is typically used and the receiver 1550 comprises or is attached to a container file generator producing a container file from input streams. Some systems operate "live," i.e. omit the recording storage

1555 and transfer coded media bitstream from the receiver 1550 directly to the decoder 1560. In some systems, only the most recent part of the recorded stream, e.g., the most recent 10-minute excerpt of the recorded stream, is maintained in the recording storage 1555, while any earlier recorded data is discarded from the recording storage 1555.

**[0222]** The coded media bitstream is transferred from the recording storage 1555 to the decoder 1560. If there are many coded media bitstreams, such as an audio stream and a video stream, associated with each other and encapsulated into a container file, a file parser (not shown in the figure) is used to decapsulate each coded media bitstream from the container file. The recording storage 1555 or a decoder 1560 may comprise the file parser, or the file parser is attached to either recording storage 1555 or the decoder 1560.

**[0223]** The coded media bitstream may be processed further by a decoder 1560, whose output is one or more uncompressed media streams. Finally, a renderer 1570 may reproduce the uncompressed media streams with a loudspeaker or a display, for example. The receiver 1550, recording storage 1555, decoder 1560, and renderer 1570 may reside in the same physical device or they may be included in separate devices.

**[0224]** Figure 17 shows a block diagram of a video coding system according to an example embodiment as a schematic block diagram of an exemplary apparatus or electronic device 50, which may incorporate a codec according to an embodiment of the invention. Figure 18 shows a layout of an apparatus according to an example embodiment. The elements of Figs. 17 and 18 will be explained next.

**[0225]** The electronic device 50 may for example be a mobile terminal or user equipment of a wireless communication system. However, it would be appreciated that embodiments of the invention may be implemented within any electronic device or apparatus which may require encoding and decoding or encoding or decoding video images.

**[0226]** The apparatus 50 may comprise a housing 30 for incorporating and protecting the device. The apparatus 50 further may comprise a display 32 in the form of a liquid crystal display. In other embodiments of the invention the display may be any suitable display technology suitable to display an image or video. The apparatus 50 may further comprise a keypad 34. In other embodiments of the invention any suitable data or user interface mechanism may be employed. For example the user interface may be implemented as a virtual keyboard or data entry system as part of a touch-sensitive display. The apparatus may comprise a microphone 36 or any suitable audio input which may be a digital or analogue signal input. The apparatus 50 may further comprise an audio output device which in embodiments of the invention may be any one of: an earpiece 38, speaker, or an analogue audio or digital audio output connection. The apparatus 50 may also comprise a battery 40 (or in other embodiments of the invention the device may be powered by any suitable mobile

energy device such as solar cell, fuel cell or clockwork generator). The apparatus may further comprise a camera 42 capable of recording or capturing images and/or video. In accordance with an embodiment, the apparatus 50 may further comprise an infrared port for short range line of sight communication to other devices. In other embodiments the apparatus 50 may further comprise any suitable short range communication solution such as for example a Bluetooth wireless connection or a USB/firewire wired connection.

**[0227]** The apparatus 50 may comprise a controller 56 or processor for controlling the apparatus 50. The controller 56 may be connected to memory 58 which in embodiments of the invention may store both data in the form of image and audio data and/or may also store instructions for implementation on the controller 56. The controller 56 may further be connected to codec circuitry 54 suitable for carrying out coding and decoding of audio and/or video data or assisting in coding and decoding carried out by the controller 56.

**[0228]** The apparatus 50 may further comprise a card reader 48 and a smart card 46, for example a UICC and UICC reader for providing user information and being suitable for providing authentication information for authentication and authorization of the user at a network.

**[0229]** The apparatus 50 may comprise radio interface circuitry 52 connected to the controller and suitable for generating wireless communication signals for example for communication with a cellular communications network, a wireless communications system or a wireless local area network. The apparatus 50 may further comprise an antenna 44 connected to the radio interface circuitry 52 for transmitting radio frequency signals generated at the radio interface circuitry 52 to other apparatus(es) and for receiving radio frequency signals from other apparatus(es).

**[0230]** In accordance with an embodiment, the apparatus 50 comprises a camera capable of recording or detecting individual frames which are then passed to the codec 54 or controller for processing. In accordance with an embodiment, the apparatus may receive the video image data for processing from another device prior to transmission and/or storage. In accordance with an embodiment, the apparatus 50 may receive either wirelessly or by a wired connection the image for coding/decoding.

**[0231]** Figure19 shows an arrangement for video coding comprising a plurality of apparatuses, networks and network elements according to an example embodiment. With respect to Figure19, an example of a system within which embodiments of the present invention can be utilized is shown. The system 10 comprises multiple communication devices which can communicate through one or more networks. The system 10 may comprise any combination of wired or wireless networks including, but not limited to a wireless cellular telephone network (such as a GSM, UMTS, CDMA network etc.), a wireless local area

network (WLAN) such as defined by any of the IEEE 802.x standards, a Bluetooth personal area network, an Ethernet local area network, a token ring local area network, a wide area network, and the Internet.

**[0232]** The system 10 may include both wired and wireless communication devices or apparatus 50 suitable for implementing embodiments of the invention. For example, the system shown in Figure 19 shows a mobile telephone network 11 and a representation of the internet 28. Connectivity to the internet 28 may include, but is not limited to, long range wireless connections, short range wireless connections, and various wired connections including, but not limited to, telephone lines, cable lines, power lines, and similar communication pathways.

**[0233]** The example communication devices shown in the system 10 may include, but are not limited to, an electronic device or apparatus 50, a combination of a personal digital assistant (PDA) and a mobile telephone 14, a PDA 16, an integrated messaging device (IMD) 18, a desktop computer 20, a notebook computer 22. The apparatus 50 may be stationary or mobile when carried by an individual who is moving. The apparatus 50 may also be located in a mode of transport including, but not limited to, a car, a truck, a taxi, a bus, a train, a boat, an airplane, a bicycle, a motorcycle or any similar suitable mode of transport.

**[0234]** Some or further apparatuses may send and receive calls and messages and communicate with service providers through a wireless connection 25 to a base station 24. The base station 24 may be connected to a network server 26 that allows communication between the mobile telephone network 11 and the internet 28. The system may include additional communication devices and communication devices of various types.

**[0235]** The communication devices may communicate using various transmission technologies including, but not limited to, code division multiple access (CDMA), global systems for mobile communications (GSM), universal mobile telecommunications system (UMTS), time divisional multiple access (TDMA), frequency division multiple access (FDMA), transmission control protocol-internet protocol (TCP-IP), short messaging service (SMS), multimedia messaging service (MMS), email, instant messaging service (IMS), Bluetooth, IEEE 802.11 and any similar wireless communication technology. A communications device involved in implementing various embodiments of the present invention may communicate using various media including, but not limited to, radio, infrared, laser, cable connections, and any suitable connection.

**[0236]** In the above, some embodiment have been described with reference to including a first cryptographic hash value (a.k.a. a first checksum) in a first container structure and indicating a first scope over which the first cryptographic hash value is computed, wherein the first scope is a subset of the data included in and/or referred to by the first container structure.

It needs to be understood that embodiments could similarly be realized when the first cryptographic hash value and/or the first scope are not included in the first container structure but in a second container structure that may e.g. be of the same hierarchy level, e.g. precede in the file order, the first container structure, or the second container structure may contain the first container structure. In some embodiments, it may be indicated with one or more syntax elements in the second container structure whether the first scope includes information in the same level (in the second container structure) or in the level below. For example, the second container structure may comprise flags to discriminate whether the boxes over which the cryptograph hash is computed is at the same containment level or a level below.

**[0237]** In the above, some embodiments have been described with reference to structures and concepts of the ISOBMFF. It needs to be understood that embodiments can similarly be realized with other container file formats, such as Matroska or its derivatives.

**[0238]** In the above, some embodiments have been described with reference to formats and concepts of DASH. It needs to be understood that embodiments can similarly be realized with other streaming and transport systems and mechanisms, such as HTTP Live Streaming (as defined in the IETF Internet Draft draft-pantos-http-live-streaming) or MMT.

**[0239]** In the above, some embodiments have been described with reference to a file. It needs to be understood that embodiments can likewise be described with reference to a file segment, such as a segment or subsegment as defined in DASH, or a resource, e.g. as understood to be received as response to an HTTP GET request.

**[0240]** Embodiments of the present invention may be implemented in software, hardware, application logic or a combination of software, hardware and application logic. In an example embodiment, the application logic, software or an instruction set is maintained on any one of various conventional computer-readable media. In the context of this document, a "computer-readable medium" may be any media or means that can contain, store, communicate, propagate or transport the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer, with one example of a computer described and depicted in Figures 17 and 18. A computer-readable medium may comprise a computer-readable storage medium that may be any media or means that can contain or store the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer.

**[0241]** If desired, the different functions discussed herein may be performed in a different order and/or concurrently with each other. Furthermore, if desired, one or more of the above-described functions may be optional or may be combined.

**[0242]** Although the above examples describe embodiments of the invention operating within a codec within an electronic device, it would be appreciated that the invention as

described below may be implemented as part of any video codec. Thus, for example, embodiments of the invention may be implemented in a video codec which may implement video coding over fixed or wired communication paths.

**[0243]** Thus, user equipment may comprise a video codec such as those described in embodiments of the invention above. It shall be appreciated that the term user equipment is intended to cover any suitable type of wireless user equipment, such as mobile telephones, portable data processing devices or portable web browsers.

**[0244]** Furthermore elements of a public land mobile network (PLMN) may also comprise video codecs as described above.

**[0245]** In general, the various embodiments of the invention may be implemented in hardware or special purpose circuits, software, logic or any combination thereof. For example, some aspects may be implemented in hardware, while other aspects may be implemented in firmware or software which may be executed by a controller, microprocessor or other computing device, although the invention is not limited thereto. While various aspects of the invention may be illustrated and described as block diagrams, flow charts, or using some other pictorial representation, it is well understood that these blocks, apparatuses, systems, techniques or methods described herein may be implemented in, as non-limiting examples, hardware, software, firmware, special purpose circuits or logic, general purpose hardware or controller or other computing devices, or some combination thereof.

**[0246]** The embodiments of this invention may be implemented by computer software executable by a data processor of the mobile device, such as in the processor entity, or by hardware, or by a combination of software and hardware. Further in this regard it should be noted that any blocks of the logic flow as in the Figures may represent program steps, or interconnected logic circuits, blocks and functions, or a combination of program steps and logic circuits, blocks and functions. The software may be stored on such physical media as memory chips, or memory blocks implemented within the processor, magnetic media such as hard disk or floppy disks, and optical media such as for example DVD and the data variants thereof, CD.

**[0247]** The various embodiments of the invention can be implemented with the help of computer program code that resides in a memory and causes the relevant apparatuses to carry out the invention. For example, a terminal device may comprise circuitry and electronics for handling, receiving and transmitting data, computer program code in a memory, and a processor that, when running the computer program code, causes the terminal device to carry out the features of an embodiment. Yet further, a network device may comprise circuitry and electronics for handling, receiving and transmitting data, computer program code in a

memory, and a processor that, when running the computer program code, causes the network device to carry out the features of an embodiment.

**[0248]** The memory may be of any type suitable to the local technical environment and may be implemented using any suitable data storage technology, such as semiconductor based memory devices, magnetic memory devices and systems, optical memory devices and systems, fixed memory and removable memory. The data processors may be of any type suitable to the local technical environment, and may include one or more of general purpose computers, special purpose computers, microprocessors, digital signal processors (DSPs) and processors based on multi core processor architecture, as non limiting examples.

**[0249]** Embodiments of the inventions may be practiced in various components such as integrated circuit modules. The design of integrated circuits is by and large a highly automated process. Complex and powerful software tools are available for converting a logic level design into a semiconductor circuit design ready to be etched and formed on a semiconductor substrate.

**[0250]** Programs, such as those provided by Synopsys Inc., of Mountain View, California and Cadence Design, of San Jose, California automatically route conductors and locate components on a semiconductor chip using well established rules of design as well as libraries of pre stored design modules. Once the design for a semiconductor circuit has been completed, the resultant design, in a standardized electronic format (e.g., Opus, GDSII, or the like) may be transmitted to a semiconductor fabrication facility or "fab" for fabrication.

**[0251]** The foregoing description has provided by way of exemplary and non-limiting examples a full and informative description of the exemplary embodiment of this invention. However, various modifications and adaptations may become apparent to those skilled in the relevant arts in view of the foregoing description, when read in conjunction with the accompanying drawings and the appended claims. However, all such and similar modifications of the teachings of this invention will still fall within the scope of this invention.

**[0252]** In the following some examples will be provided.

**[0253]** There is provided a method comprising:

receiving a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

obtaining information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

deriving a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

determining the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the first scope being valid, processing the data within the first scope.

**[0254]** In an embodiment the container file further includes or refers to a second checksum, and the method further comprises:

parsing from the file or concluding a second scope for the second checksum, at least a part of the second scope being optional for the processing of the at least one media data unit;

determining the validity of the data within the second scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the second scope being valid, processing the data within the second scope.

**[0255]** In an embodiment the method comprises:

obtaining information from the container file of an algorithm by which the first checksum has been calculated.

**[0256]** In an embodiment the container file comprises encoded information on two or more hierarchy levels, wherein the method comprises:

obtaining or deriving from the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

**[0257]** In an embodiment the method comprises:

obtaining or deriving from the container file a switch code; and

using the value of the switch code to determine the hierarchy level of the first scope.

**[0258]** In an embodiment the method comprises:

determining that the first scope is within the same hierarchy level than the switch code, if the value of switch code is a first value; or

determining that the first scope is one level lower of the hierarchy than the switch code, if the value of switch code is a second value; or

determining that the first scope contains certain types of sample information, if the value of switch code is a third value.

**[0259]** In an embodiment the method comprises:

obtaining or deriving from the container file a flag; and

using the flag to determine the hierarchy level of the first scope.

**[0260]** In an embodiment the method comprises:

determining that there is only one checksum and first scope, if the flag has a first flag value; or

determining that there is more than one checksum and corresponding more than one first scope, if the flag has a second flag value.

**[0261]** In an embodiment the method comprises:

obtaining the information of the first scope and the first checksum from an integrity header box of a data integrity box of an ISOBMFF file.

**[0262]** In a second example, there is provided an apparatus comprising at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

receive a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

obtain information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

derive a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

determine the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the first scope being valid, process the data within the first scope.

**[0263]** In an embodiment of the apparatus the container file further includes or refers to a second checksum, and said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

parse from the file or concluding a second scope for the second checksum, at least a part of the second scope being optional for the processing of the at least one media data unit;

determine the validity of the data within the second scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the second scope being valid, process the data within the second scope.

**[0264]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

obtain information from the container file of an algorithm by which the first checksum has been calculated.

**[0265]** In an embodiment of the apparatus the container file comprises encoded information on two or more hierarchy levels, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

obtain or derive from the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

**[0266]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

obtain or derive from the container file a switch code; and

use the value of the switch code to determine the hierarchy level of the first scope.

**[0267]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

determine that the first scope is within the same hierarchy level than the switch code, if the value of switch code is a first value; or

determine that the first scope is one level lower of the hierarchy than the switch code, if the value of switch code is a second value; or

determine that the first scope contains certain types of sample information, if the value of switch code is a third value.

**[0268]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

obtain or derive from the container file a flag; and

use the flag to determine the hierarchy level of the first scope.

**[0269]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

determine that there is only one checksum and first scope, if the flag has a first flag value; or

determine that there is more than one checksum and corresponding more than one first scope, if the flag has a second flag value.

**[0270]** In an embodiment the method comprises:

obtaining the information of the first scope and the first checksum from an integrity header box of a data integrity box of an ISOBMFF file.

**[0271]** In a third example, there is provided a computer program product embodied on a non-transitory computer readable medium, comprising computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

receive a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

obtain information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

derive a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

determine the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the first scope being valid, process the data within the first scope.

**[0272]** In an embodiment of the computer program product the container file further includes or refers to a second checksum, wherein the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

parse from the file or concluding a second scope for the second checksum, at least a part of the second scope being optional for the processing of the at least one media data unit;

determine the validity of the data within the second scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the second scope being valid, process the data within the second scope.

**[0273]** In an embodiment of the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

obtain information from the container file of an algorithm by which the first checksum has been calculated.

**[0274]** In an embodiment of the computer program product the container file comprises encoded information on two or more hierarchy levels, wherein the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

obtain or derive from the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

**[0275]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

obtain or derive from the container file a switch code; and

use the value of the switch code to determine the hierarchy level of the first scope.

**[0276]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

determine that the first scope is within the same hierarchy level than the switch code, if the value of switch code is a first value; or

determine that the first scope is one level lower of the hierarchy than the switch code, if the value of switch code is a second value; or

determine that the first scope contains certain types of sample information, if the value of switch code is a third value.

**[0277]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

obtain or derive from the container file a flag; and

use the flag to determine the hierarchy level of the first scope.

**[0278]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

determine that there is only one checksum and first scope, if the flag has a first flag value; or

determine that there is more than one checksum and corresponding more than one first scope, if the flag has a second flag value.

**[0279]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

obtain the information of the first scope and the first checksum from an integrity header box of a data integrity box of an ISOBMFF file.

**[0280]** In a fourth example, there is provided a method comprising:

forming a container file according to a first format;

including the container file at least one meta data unit,

including the container file or referring in the container file to at least one media data unit and a first checksum; and

**[0281]** including the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit. In an embodiment the method comprises:

forming a second checksum from a second scope, at least a part of the second scope being optional for the processing of the at least one media data unit.

**[0282]** In an embodiment the method comprises:

including into the container file information of an algorithm by which the first checksum has been calculated.

**[0283]** In an embodiment of the method the container file comprises encoded information on two or more hierarchy levels, wherein the method comprises:

including into the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

**[0284]** In an embodiment the method comprises:

including into the container file a switch code indicative of the hierarchy level of the first scope.

**[0285]** In an embodiment the method comprises:

setting the switch code to a first value, if the first scope is within the same hierarchy level than the switch code; or

setting the switch code to a second value, if the first scope is one level lower of the hierarchy than the switch code; or

setting the switch code to a third value, if the first scope contains certain types of sample information.

**[0286]** In an embodiment the method comprises:

including into the container file a flag indicative of the hierarchy level of the first scope.

**[0287]** In an embodiment the method comprises:

setting the flag into a first flag value, if there is only one checksum and first scope; or

setting the flag into a second flag value, if there is more than one checksum and more than one corresponding first scope.

**[0288]** In an embodiment the method comprises:

forming the information of the first scope and the first checksum to an integrity header box of a data integrity box of an ISOBMFF file.

**[0289]** In a fifth example, there is provided an apparatus comprising at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

form a container file according to a first format;

include the container file at least one meta data unit,

include the container file or referring in the container file to at least one media data unit and a first checksum; and

include the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.

**[0290]** In an embodiment of the apparatus the container file further includes or refers to a second checksum, and said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

form a second checksum from a second scope, at least a part of the second scope being optional for the processing of the at least one media data unit.

**[0291]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

include into the container file information of an algorithm by which the first checksum has been calculated.

**[0292]** In an embodiment of the apparatus the container file comprises encoded information on two or more hierarchy levels, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

include into the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

**[0293]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

include into the container file a switch code indicative of the hierarchy level of the first scope.

**[0294]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

set the switch code to a first value, if the first scope is within the same hierarchy level than the switch code; or

set the switch code to a second value, if the first scope is one level lower of the hierarchy than the switch code; or

set the switch code to a third value, if the first scope contains certain types of sample information.

**[0295]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

include into the container file a flag indicative of the hierarchy level of the first scope.

**[0296]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

set the flag into a first flag value, if there is only one checksum and first scope; or

set the flag into a second flag value, if there is more than one checksum and more than one corresponding first scope.

**[0297]** In an embodiment of the apparatus said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

form the information of the first scope and the first checksum to an integrity header box of a data integrity box of an ISOBMFF file.

**[0298]** In a sixth example, there is provided a computer program product embodied on a non-transitory computer readable medium, comprising computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

form a container file according to a first format;

include the container file at least one meta data unit,

include the container file or referring in the container file to at least one media data unit and a first checksum; and

include the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.

**[0299]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

form a second checksum from a second scope, at least a part of the second scope being optional for the processing of the at least one media data unit.

**[0300]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

include into the container file information of an algorithm by which the first checksum has been calculated.

**[0301]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

include into the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

**[0302]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

include into the container file a switch code indicative of the hierarchy level of the first scope.

**[0303]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

set the switch code to a first value, if the first scope is within the same hierarchy level than the switch code; or

set the switch code to a second value, if the first scope is one level lower of the hierarchy than the switch code; or

set the switch code to a third value, if the first scope contains certain types of sample information.

**[0304]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

include into the container file a flag indicative of the hierarchy level of the first scope.

**[0305]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

set the flag into a first flag value, if there is only one checksum and first scope; or

set the flag into a second flag value, if there is more than one checksum and more than one corresponding first scope.

**[0306]** In an embodiment the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

**[0307]** form the information of the first scope and the first checksum to an integrity header box of a data integrity box of an ISOBMFF file. In a seventh example, there is provided an apparatus configured to perform the method of the first example.

**[0308]** In an eighth example, there is provided an apparatus configured to perform the method of the fourth example.

**[0309]** In a ninth example, there is provided an apparatus comprising:

means for receiving a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

means for obtaining information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

means for deriving a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

means for determining the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

means for processing the data within the first scope, subject to the data within the first scope being valid.

**[0310]** In a tenth example, there is provided an apparatus comprising:

means for forming a container file according to a first format;

means for including the container file at least one meta data unit,

means for including the container file or referring in the container file to at least one media data unit and a first checksum; and

means for including the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.

**CLAIMS**

1. A method comprising:
  - receiving a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;
  - obtaining information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;
  - deriving a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;
  - determining the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and
  - subject to the data within the first scope being valid, processing the data within the first scope.
2. The method as claimed in Claim 1, wherein the container file further includes or refers to a second checksum, and the method further comprises:
  - parsing from the file or concluding a second scope for the second checksum, at least a part of the second scope being optional for the processing of the at least one media data unit;
  - determining the validity of the data within the second scope on the basis of the first checksum and the first reproduced checksum; and
  - subject to the data within the second scope being valid, processing the data within the second scope.
3. The method as claimed in Claims 1 and 2 further comprising:
  - obtaining information from the container file of an algorithm by which the first checksum has been calculated.
4. The method as claimed in Claims 1 to 3, wherein the container file comprises encoded information on two or more hierarchy levels, and wherein the method further comprises:
  - obtaining or deriving from the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.
5. The method as claimed in Claim 4 further comprising:
  - obtaining or deriving from the container file a switch code; and

using the value of the switch code to determine the hierarchy level of the first scope.

6. The method as claimed in Claim 5 further comprising:
  - determining that the first scope is within the same hierarchy level than the switch code, if the value of switch code is a first value; or
  - determining that the first scope is one level lower of the hierarchy than the switch code, if the value of switch code is a second value; or
  - determining that the first scope contains certain types of sample information, if the value of switch code is a third value.
7. The method as claimed in Claims 4 to 6 further comprising:
  - obtaining or deriving from the container file a flag; and
  - using the flag to determine the hierarchy level of the first scope.
8. The method as claimed in Claim 7 further comprising:
  - determining that there is only one checksum and first scope, if the flag has a first flag value; or
  - determining that there is more than one checksum and corresponding more than one first scope, if the flag has a second flag value.
9. The method as claimed in Claims 1 to 9 further comprising:
  - obtaining the information of the first scope and the first checksum from an integrity header box of a data integrity box of an ISOBMFF file.
10. An apparatus comprising at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:
  - receive a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;
  - obtain information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;
  - derive a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;
  - determine the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the first scope being valid, process the data within the first scope.

11. The apparatus as claimed in claim 10, wherein the container file further includes or refers to a second checksum, and said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

parse from the file or concluding a second scope for the second checksum, at least a part of the second scope being optional for the processing of the at least one media data unit;

determine the validity of the data within the second scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the second scope being valid, process the data within the second scope.

12. The apparatus as claimed in Claims 10 and 11, wherein the at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

obtain information from the container file of an algorithm by which the first checksum has been calculated.

13. The apparatus as claimed in Claims 10 to 12, wherein the container file comprises encoded information on two or more hierarchy levels, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

obtain or derive from the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

14. The apparatus as claimed in Claim 13, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

obtain or derive from the container file a switch code; and

use the value of the switch code to determine the hierarchy level of the first scope.

15. The apparatus as claimed in Claim 14, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

determine that the first scope is within the same hierarchy level than the switch code, if the value of switch code is a first value; or

determine that the first scope is one level lower of the hierarchy than the switch code, if the value of switch code is a second value; or

determine that the first scope contains certain types of sample information, if the value of switch code is a third value.

16. The apparatus as claimed in Claims 13 to 15, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

obtain or derive from the container file a flag; and

use the flag to determine the hierarchy level of the first scope.

17. The apparatus as claimed in Claim 16, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

determine that there is only one checksum and first scope, if the flag has a first flag value; or

determine that there is more than one checksum and corresponding more than one first scope, if the flag has a second flag value.

18. The apparatus as claimed in Claims 10 to 17, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

obtain the information of the first scope and the first checksum from an integrity header box of a data integrity box of an ISOBMFF file.

19. A computer program product embodied on a non-transitory computer readable medium, comprising computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

receive a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

obtain information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

derive a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

determine the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the first scope being valid, process the data within the first scope.

20. The computer program product as claimed in Claim 19, wherein the container file further includes or refers to a second checksum, wherein the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

parse from the file or concluding a second scope for the second checksum, at least a part of the second scope being optional for the processing of the at least one media data unit;

determine the validity of the data within the second scope on the basis of the first checksum and the first reproduced checksum; and

subject to the data within the second scope being valid, process the data within the second scope.

21. The computer program product as claimed in Claims 19 and 20 comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

obtain information from the container file of an algorithm by which the first checksum has been calculated.

22. The computer program product as claimed in Claims 19 to 21, wherein the container file comprises encoded information on two or more hierarchy levels, wherein the computer program product comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

obtain or derive from the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

23. The computer program product as claimed in Claim 22 comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

- obtain or derive from the container file a switch code; and
- use the value of the switch code to determine the hierarchy level of the first scope.

24. The computer program product as claimed in Claim 23 comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

- determine that the first scope is within the same hierarchy level than the switch code, if the value of switch code is a first value; or
- determine that the first scope is one level lower of the hierarchy than the switch code, if the value of switch code is a second value; or
- determine that the first scope contains certain types of sample information, if the value of switch code is a third value.

25. The computer program product as claimed in Claims 22 to 24 comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

- obtain or derive from the container file a flag; and
- use the flag to determine the hierarchy level of the first scope.

26. The computer program product as claimed in Claim 25 comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

- determine that there is only one checksum and first scope, if the flag has a first flag value; or
- determine that there is more than one checksum and corresponding more than one first scope, if the flag has a second flag value.

27. The computer program product as claimed in Claims 19 to 26 comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

- obtain the information of the first scope and the first checksum from an integrity header box of a data integrity box of an ISOBMFF file.

28. A method comprising:  
forming a container file according to a first format;  
including the container file at least one meta data unit,  
including the container file or referring in the container file to at least one media data unit and a first checksum; and  
including the container file information of a first scope for the first checksum,  
wherein the first scope is a subset of the container file and/or the at least one media data unit,  
the first scope being for processing of the at least one media data unit.
29. The method as claimed in Claim 28, further comprising:  
forming a second checksum from a second scope, at least a part of the second scope being optional for the processing of the at least one media data unit.
30. The method as claimed in Claims 28 and 29, further comprises:  
including into the container file information of an algorithm by which the first checksum has been calculated.
31. The method as claimed in Claims 28 to 30 wherein the container file comprises encoded information on two or more hierarchy levels and wherein the method further comprises:  
including into the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.
32. The method as claimed in Claim 31, further comprising:  
including into the container file a switch code indicative of the hierarchy level of the first scope.
33. The method as claimed in Claim 32 further comprising:  
setting the switch code to a first value, if the first scope is within the same hierarchy level than the switch code; or  
setting the switch code to a second value, if the first scope is one level lower of the hierarchy than the switch code; or  
setting the switch code to a third value, if the first scope contains certain types of sample information.

34. The method as claimed in Claims 31 to 33 further comprising:  
including into the container file a flag indicative of the hierarchy level of the first scope.
35. The method as claimed in Claim 34, further comprising:  
setting the flag into a first flag value, if there is only one checksum and first scope;  
or  
setting the flag into a second flag value, if there is more than one checksum and more than one corresponding first scope.
36. The method as claimed in Claims 28 to 35 further comprising:  
forming the information of the first scope and the first checksum to an integrity header box of a data integrity box of an ISOBMFF file.
37. An apparatus comprising at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:  
form a container file according to a first format;  
include the container file at least one meta data unit,  
include the container file or referring in the container file to at least one media data unit and a first checksum; and  
include the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.
38. The apparatus as claimed in Claim 37, wherein the container file further includes or refers to a second checksum, and said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:  
form a second checksum from a second scope, at least a part of the second scope being optional for the processing of the at least one media data unit.
39. The apparatus as claimed in Claims 37 and 38, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:  
include into the container file information of an algorithm by which the first checksum has been calculated.

40. The apparatus as claimed in Claims 37 to 39, wherein the container file comprises encoded information on two or more hierarchy levels, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

include into the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

41. The apparatus as claimed in Claim 40, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

include into the container file a switch code indicative of the hierarchy level of the first scope.

42. The apparatus as claimed in Claim 41, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

set the switch code to a first value, if the first scope is within the same hierarchy level than the switch code; or

set the switch code to a second value, if the first scope is one level lower of the hierarchy than the switch code; or

set the switch code to a third value, if the first scope contains certain types of sample information.

43. The apparatus as claimed in Claims 40 to 42, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

include into the container file a flag indicative of the hierarchy level of the first scope.

44. The apparatus as claimed in 43, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

set the flag into a first flag value, if there is only one checksum and first scope; or

set the flag into a second flag value, if there is more than one checksum and more than one corresponding first scope.

45. The apparatus as claimed in Claims 37 to 44, wherein said at least one memory stored with code thereon, which when executed by said at least one processor, causes an apparatus to perform at least the following:

form the information of the first scope and the first checksum to an integrity header box of a data integrity box of an ISOBMFF file.

46. A Computer program product embodied on a non-transitory computer readable medium, comprising computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

form a container file according to a first format;

include the container file at least one meta data unit,

include the container file or referring in the container file to at least one media data unit and a first checksum; and

include the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.

47. The computer program product as claimed in Claim 46, comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

form a second checksum from a second scope, at least a part of the second scope being optional for the processing of the at least one media data unit.

48. The computer program product as claimed in Claims 46 and 47, comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

include into the container file information of an algorithm by which the first checksum has been calculated.

49. The computer program product as claimed in Claim 46 to 48, comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

include into the container file information whether the first scope is on the same hierarchy level than the information on the first scope or on a lower level.

50. The computer program product as claimed in Claim 49, comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

include into the container file a switch code indicative of the hierarchy level of the first scope.

51. The computer program product as claimed in Claim 50, comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

set the switch code to a first value, if the first scope is within the same hierarchy level than the switch code; or

set the switch code to a second value, if the first scope is one level lower of the hierarchy than the switch code; or

set the switch code to a third value, if the first scope contains certain types of sample information.

52. The computer program product as claimed in Claims 49 to 51, comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

include into the container file a flag indicative of the hierarchy level of the first scope.

53. The computer program product as claimed in Claim 52, comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

set the flag into a first flag value, if there is only one checksum and first scope; or

set the flag into a second flag value, if there is more than one checksum and more than one corresponding first scope.

54. The computer program product as claimed in Claims 46 to 53 comprises computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:

form the information of the first scope and the first checksum to an integrity header box of a data integrity box of an ISOBMFF file.

55. An apparatus configured to perform the method according to any of Claims 1 to 9.

56. An apparatus configured to perform the method according to any of Claims 28 to 36.

57. An apparatus comprising:

means for receiving a container file according to a first format, the container file including at least one meta data unit, the container file including or referring to at least one media data unit, and the container file including or referring to a first checksum;

means for obtaining information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit;

means for deriving a first reproduced checksum from the container file and/or the at least one media data unit on the basis of the first scope;

means for determining the validity of the data within the first scope on the basis of the first checksum and the first reproduced checksum; and

means for processing the data within the first scope, subject to the data within the first scope being valid.

58. An apparatus comprising:

means for forming a container file according to a first format;

means for including the container file at least one meta data unit,

means for including the container file or referring in the container file to at least one media data unit and a first checksum; and

means for including the container file information of a first scope for the first checksum, wherein the first scope is a subset of the container file and/or the at least one media data unit, the first scope being for processing of the at least one media data unit.

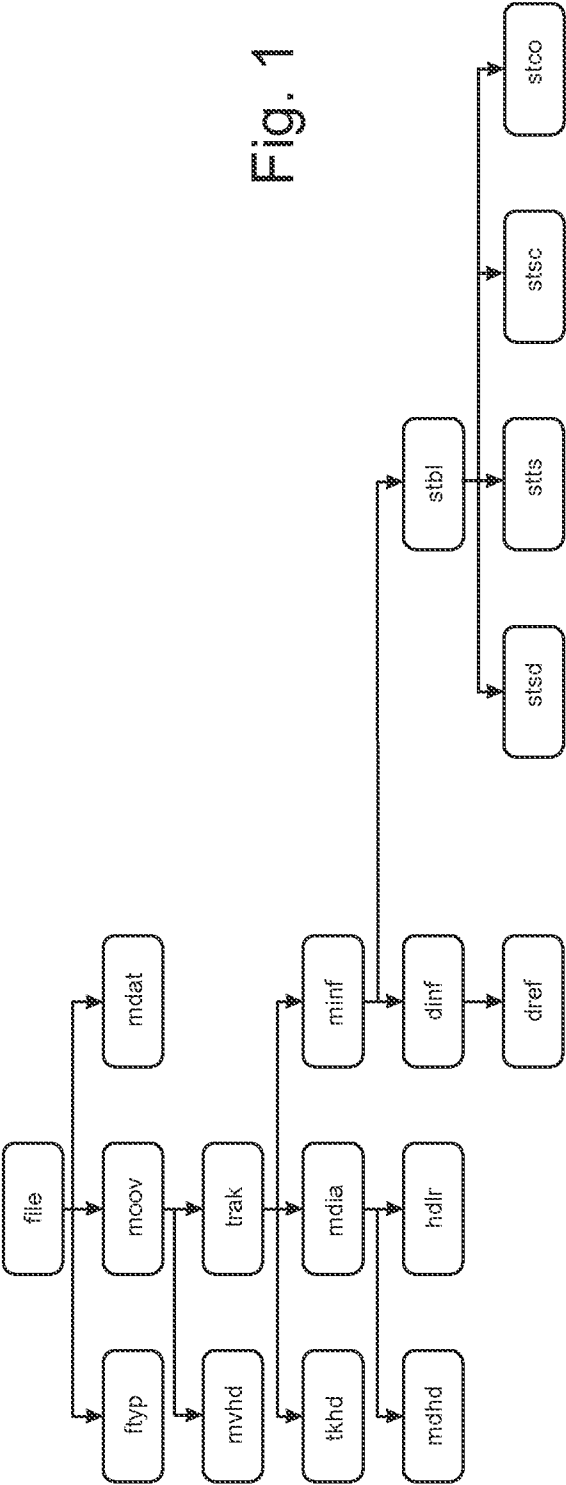


Fig. 1

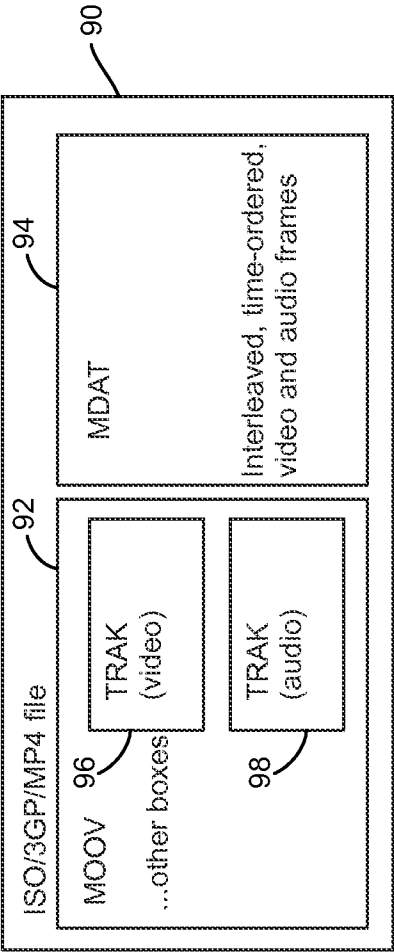


Fig. 2

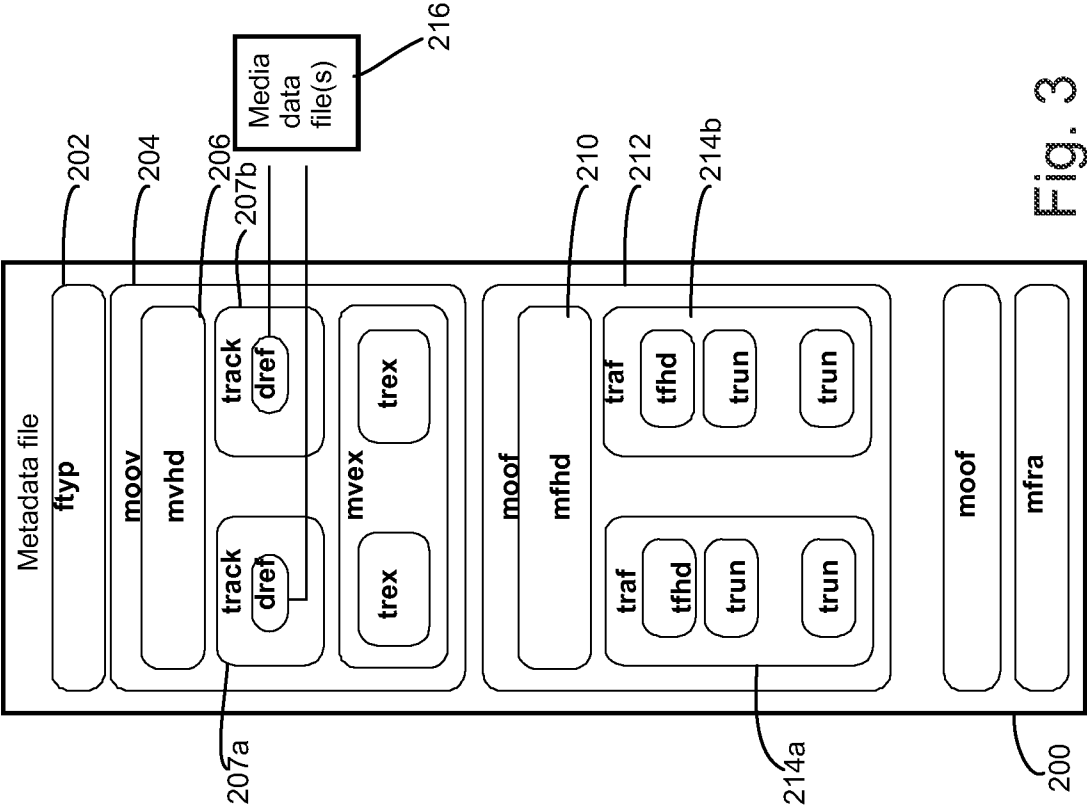


Fig. 3

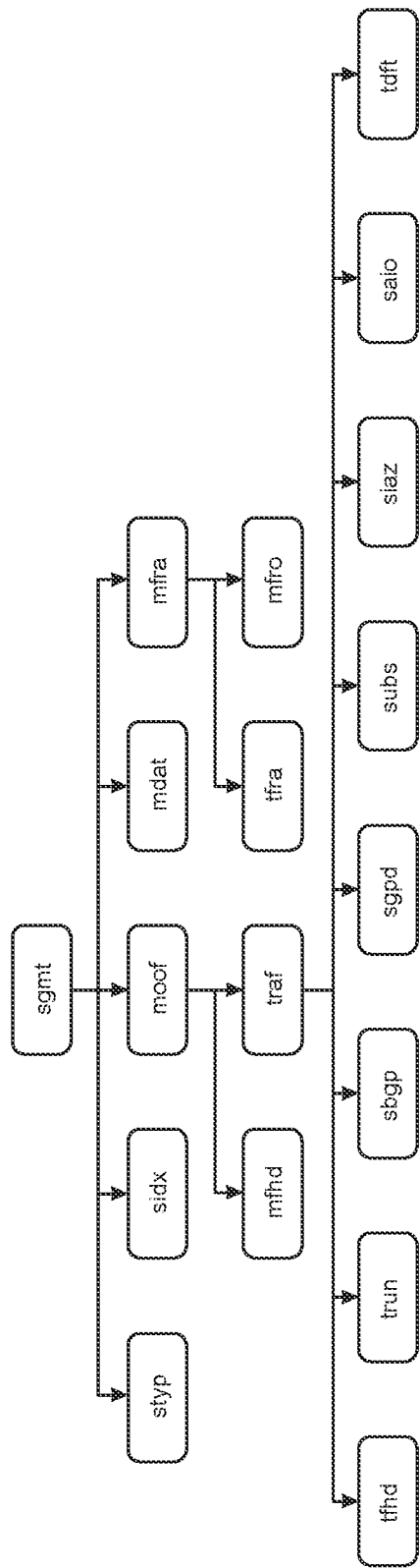


Fig. 4

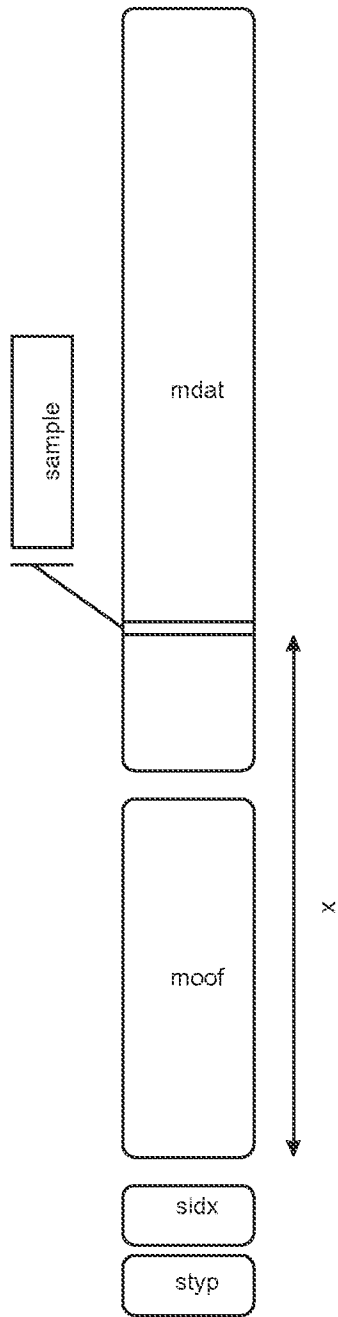


Fig. 5

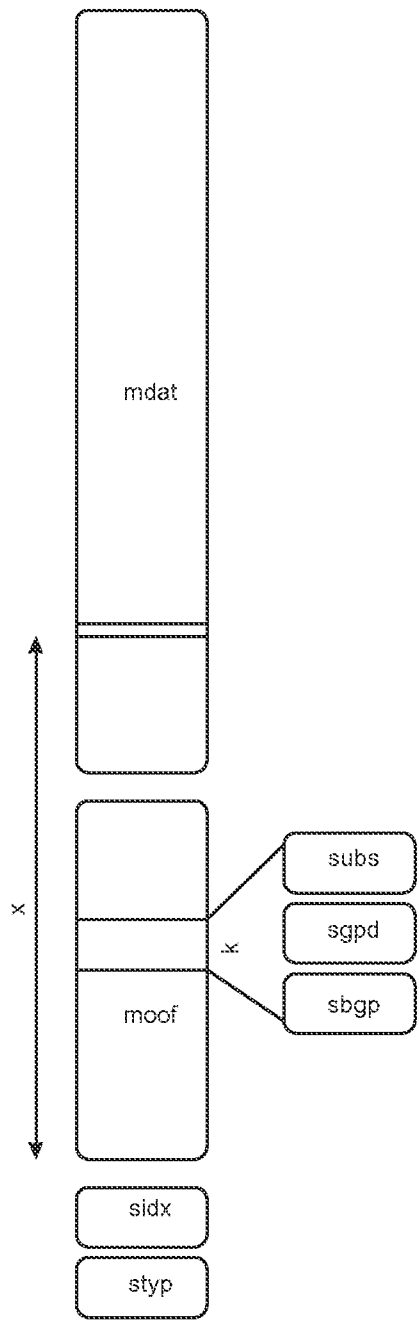


Fig. 6

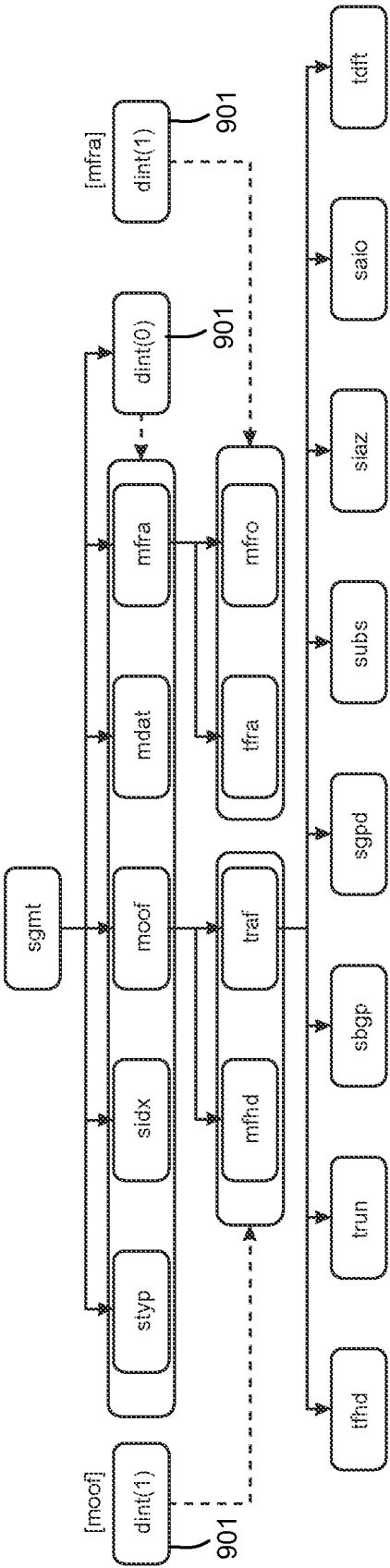


Fig. 7

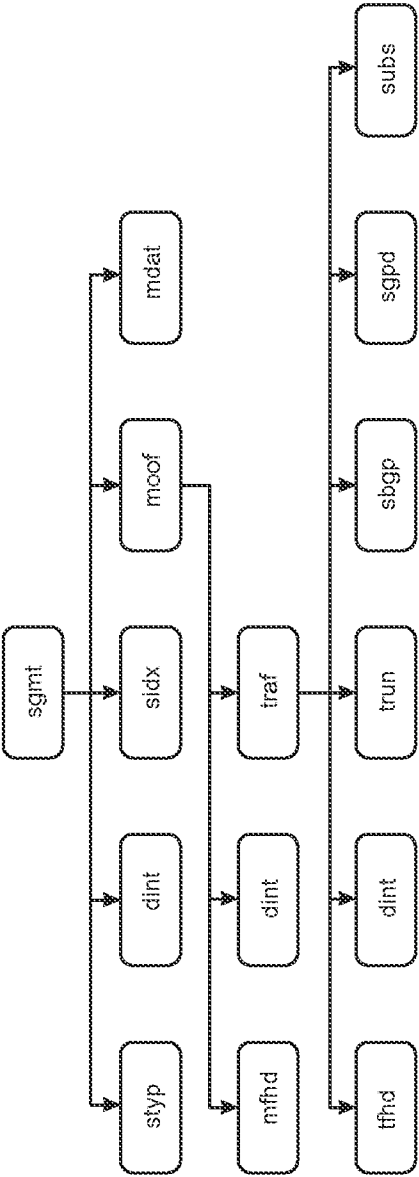


Fig. 8

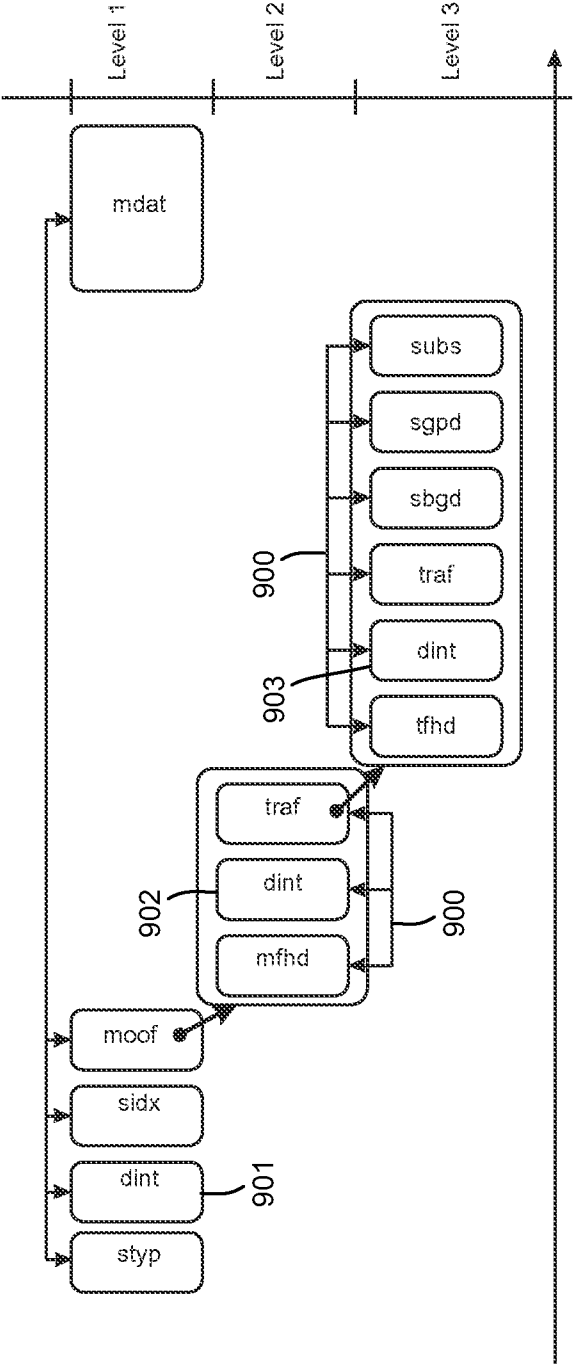


Fig. 9

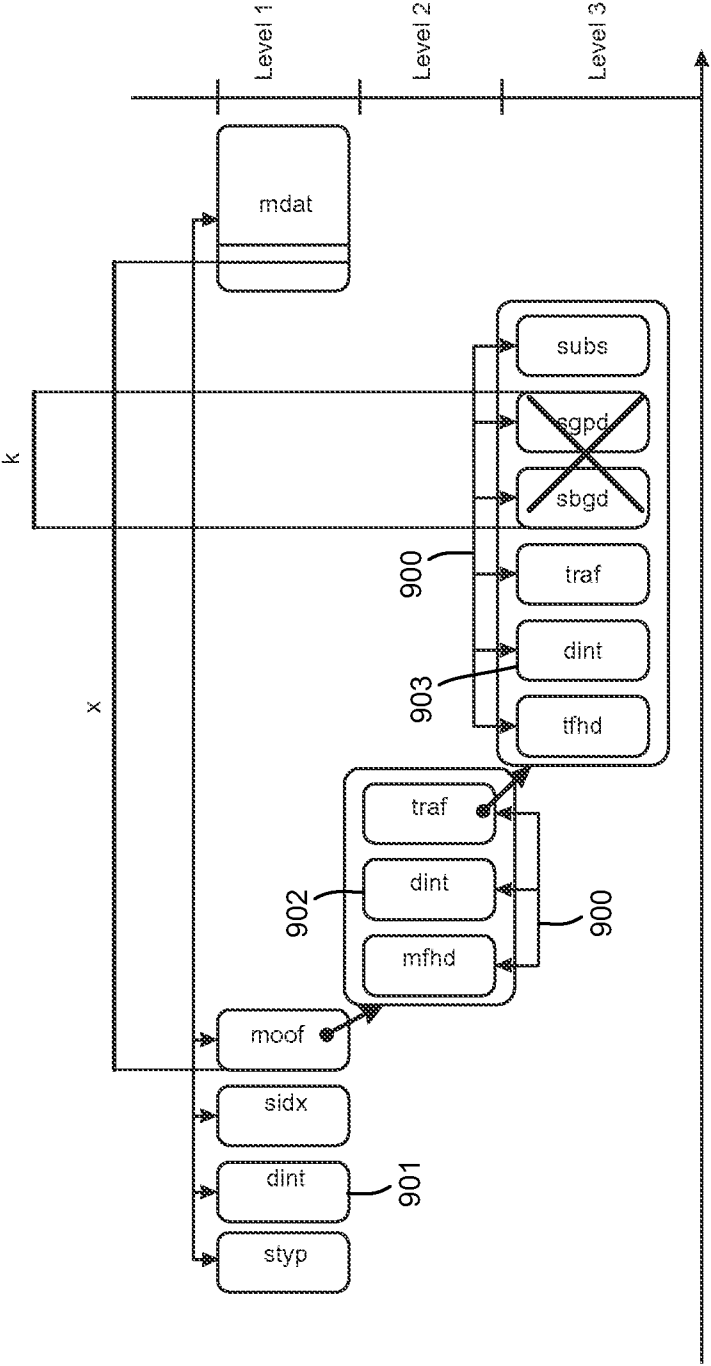


Fig. 10

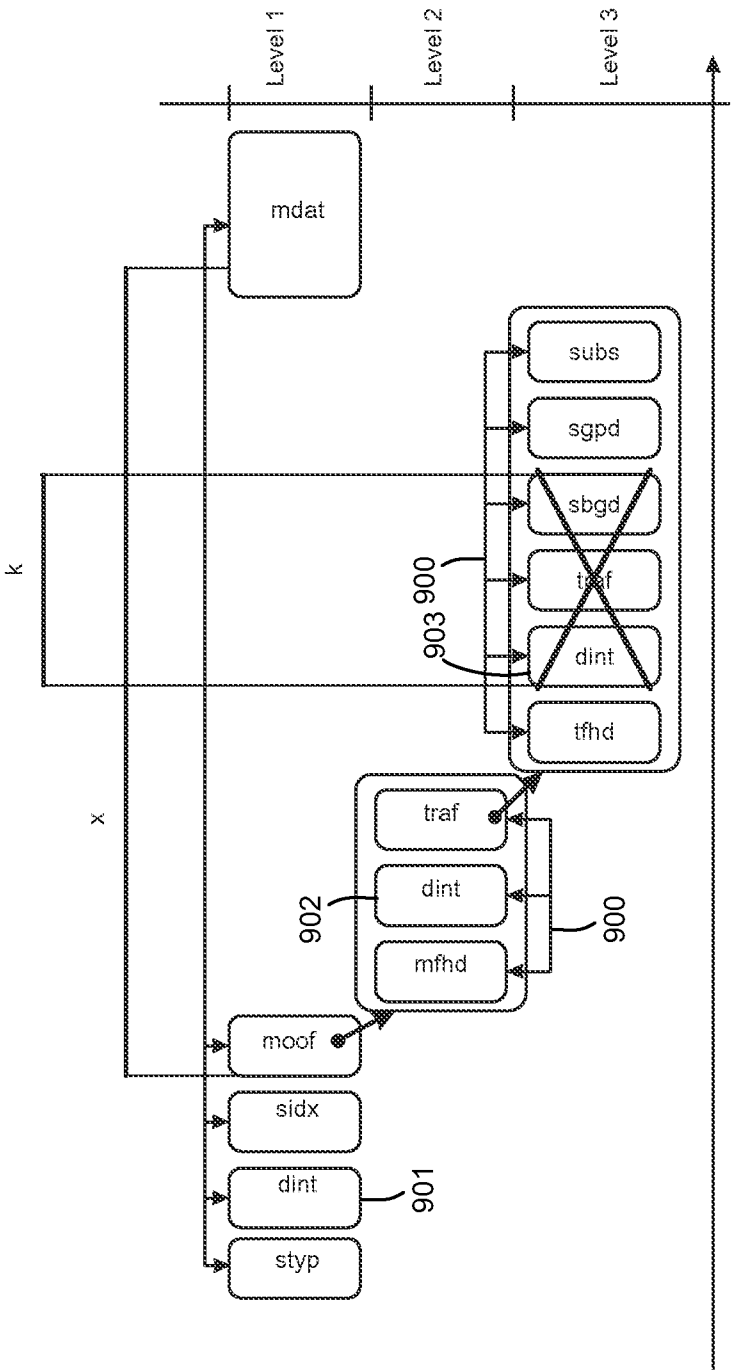
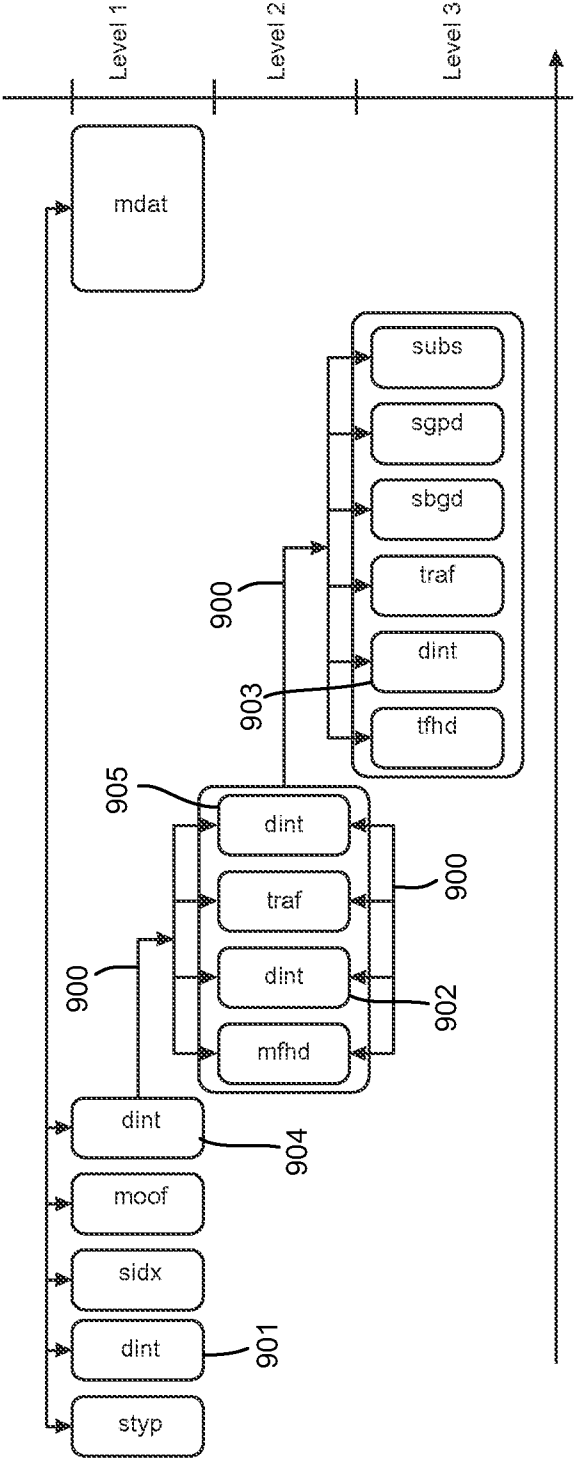


Fig. 11

Fig. 12



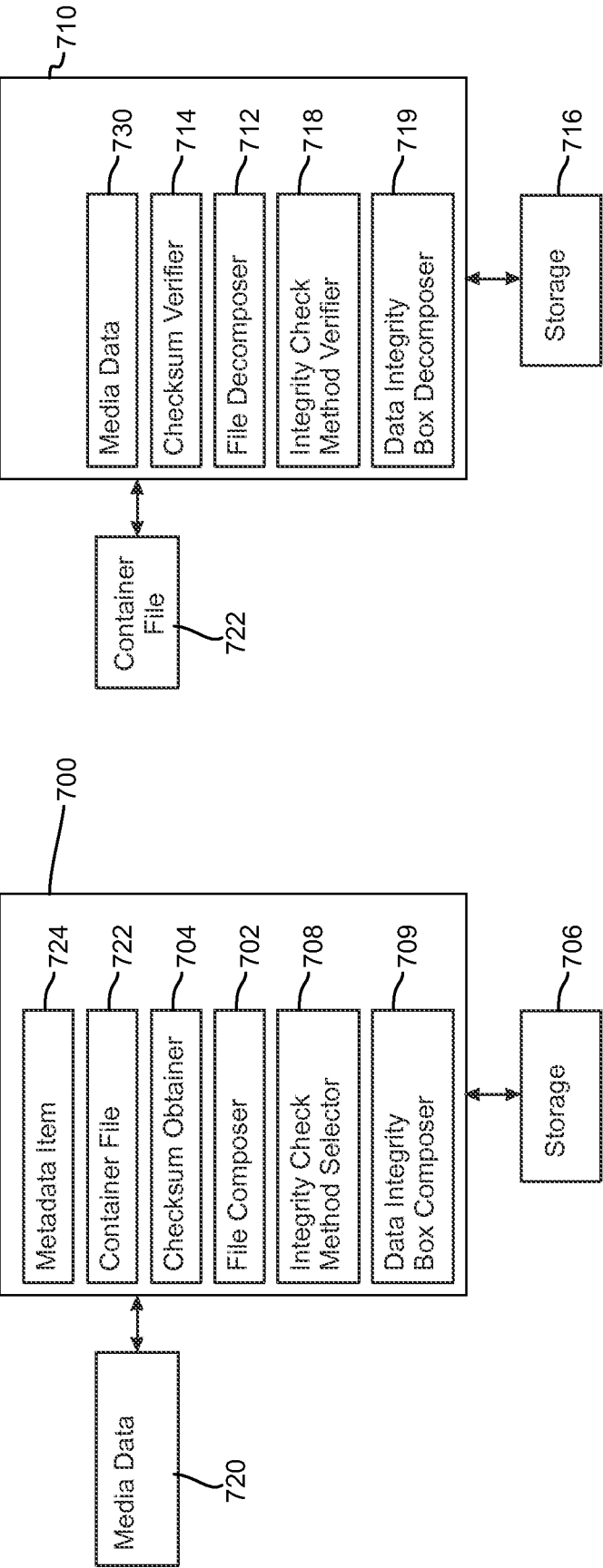


Fig. 13a

Fig. 13b

10/15

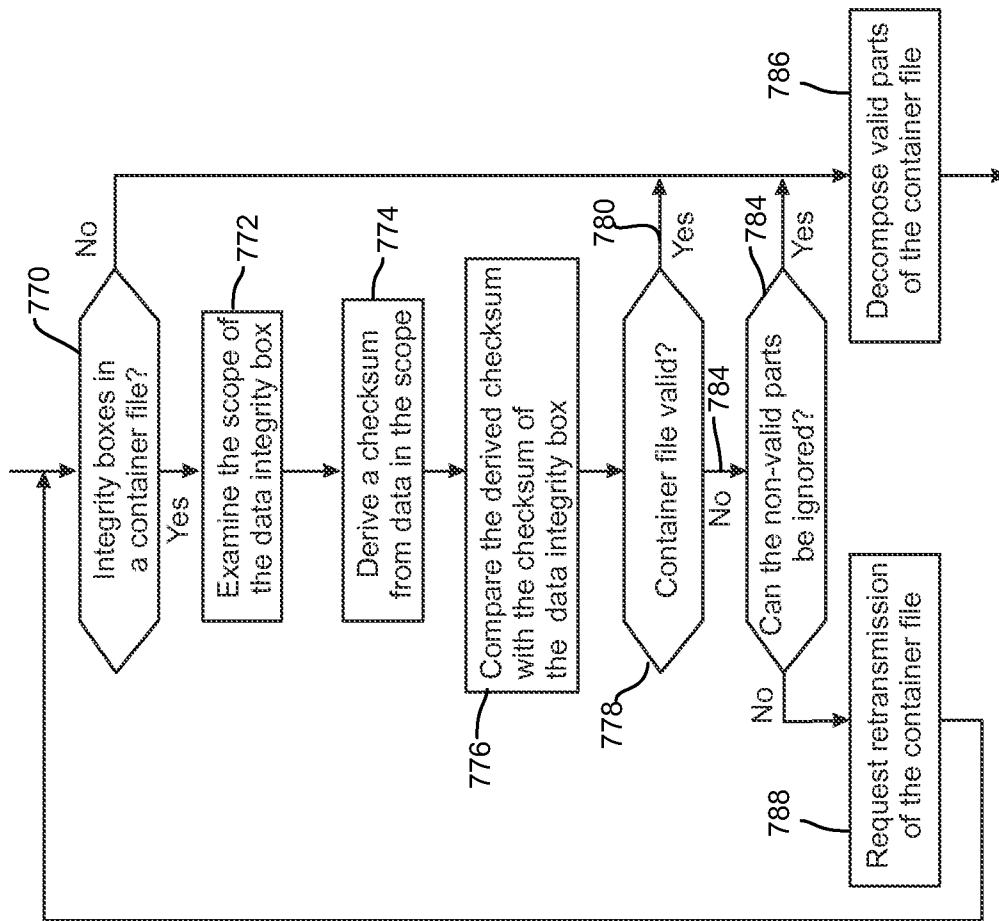


Fig. 13d

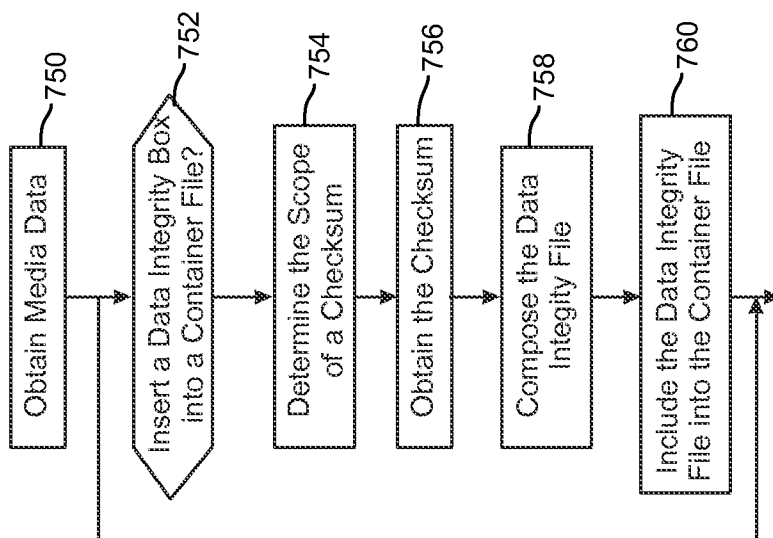


Fig. 13c

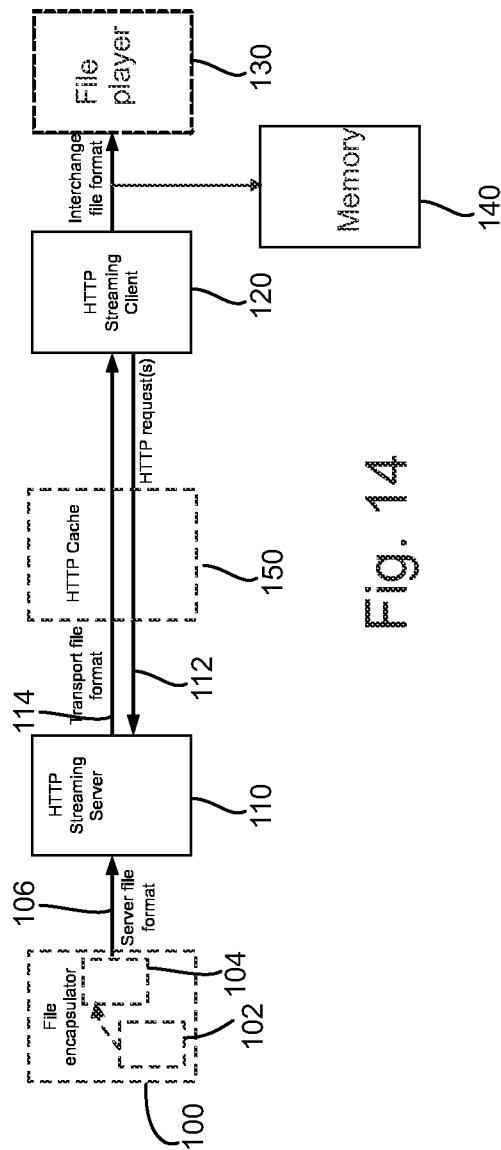


Fig. 14

12/15

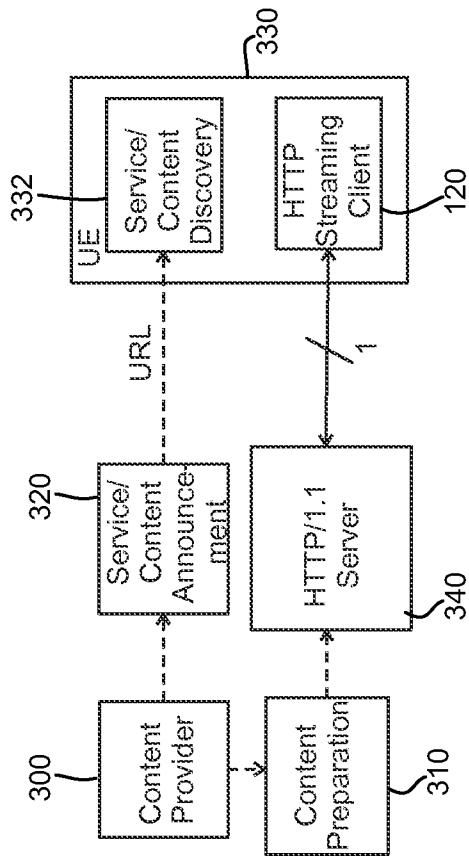


Fig. 15a

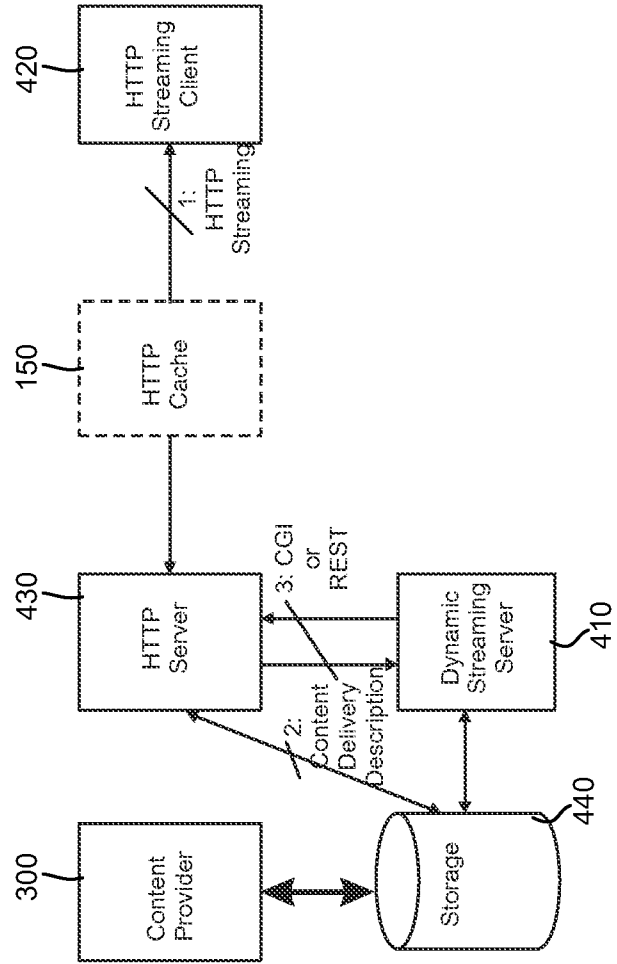


Fig. 15b

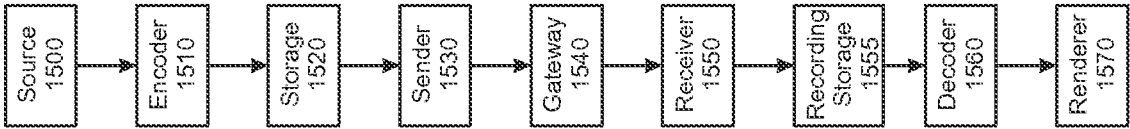


Fig. 16

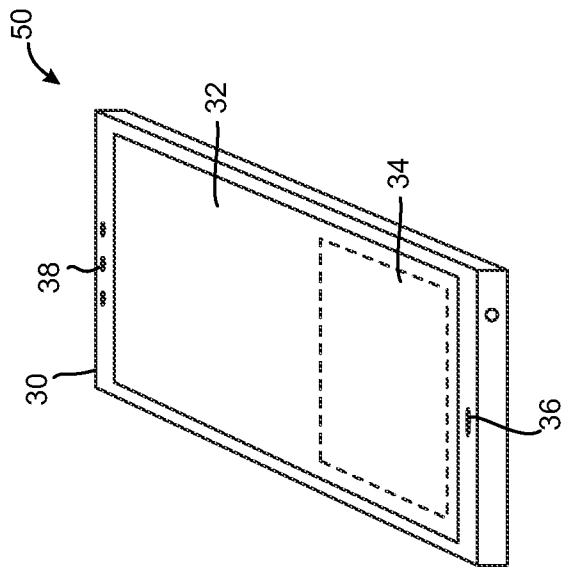


Fig. 18

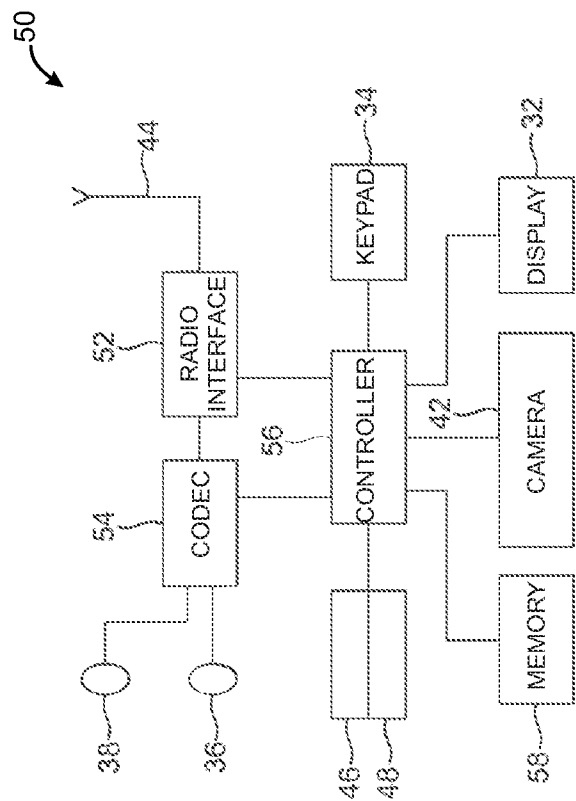


Fig. 17

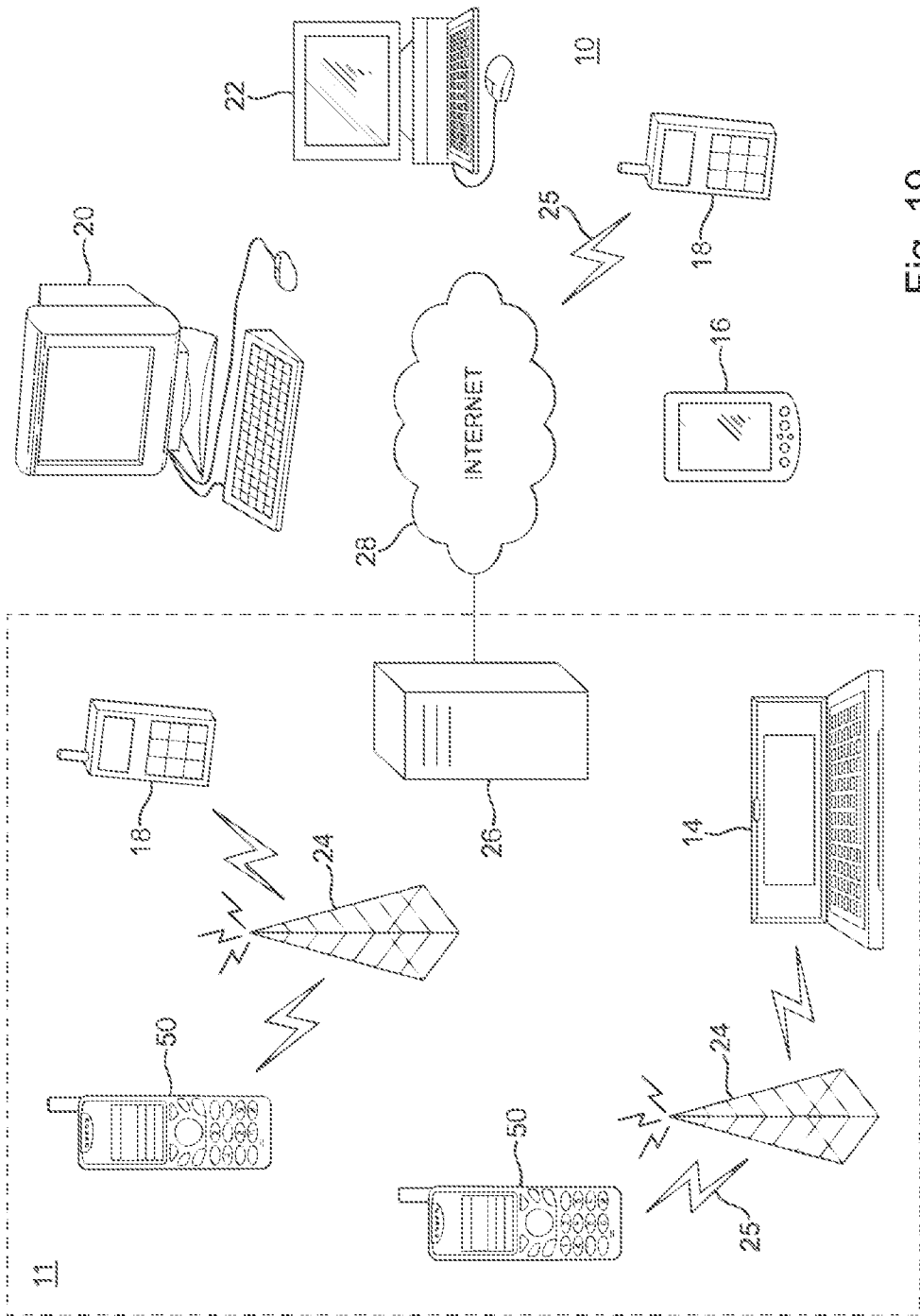


Fig. 19

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2015/050885

**A. CLASSIFICATION OF SUBJECT MATTER**

See extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F, H04L, H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched  
FI, SE, NO, DK

Electronic data base consulted during the international search (name of data base, and, where practicable, search terms used)

EPO-Internal, WPIAP, NPL, INSPEC, XP3GPP, XPAIP, XPESP, XPESP2, XPETSI, XPI3E, XPIEE, XPIETF, XPIOP, XPIPCOM, XPJPEG, XPOAC, XPRD, XPTK, COMPDX, Internet

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                     | Relevant to claim No.                                  |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| X         | US 2002162076 A1 (TALAGALA NISHA D [US] et al.)<br>31 October 2002 (31.10.2002)<br>the whole document, in particular paragraphs [0016]-[0018], [0025]-[0026], [0052], [0054], [0059], [0072]-[0084]; Figs. 7A, 7B, 8-9 | 1-58                                                   |
| X         | US 2002023249 A1 (TERNULLO NOAH J [US] et al.)<br>21 February 2002 (21.02.2002)<br>abstract; paragraphs [0020]-[0024], [0061], [0072], [0092], [0140]; Fig. 4A; claims 1, 5-7                                          | 1-58                                                   |
| X         | US 2003188231 A1 (CRONCE PAUL A [US])<br>02 October 2003 (02.10.2003)<br>abstract; paragraphs [0004]-[0006], [0026]-[0028], [0030]-[0035], [0038], [0048], [0053]-[0054]; Figs. 1, 3A, 3B, 4B, 6, 8-9; claims 1, 4     | 1-3, 10-12,<br>19-21, 28-30,<br>37-39, 46-48,<br>55-58 |



Further documents are listed in the continuation of Box C.



See patent family annex.

|                                                                                                                                                                         |                                                                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| * Special categories of cited documents:                                                                                                                                | "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention                                              |
| "A" document defining the general state of the art which is not considered to be of particular relevance                                                                | "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone                                                                     |
| "E" earlier application or patent but published on or after the international filing date                                                                               | "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art |
| "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) | "&" document member of the same patent family                                                                                                                                                                                                    |
| "O" document referring to an oral disclosure, use, exhibition or other means                                                                                            |                                                                                                                                                                                                                                                  |
| "P" document published prior to the international filing date but later than the priority date claimed                                                                  |                                                                                                                                                                                                                                                  |

Date of the actual completion of the international search

17 April 2016 (17.04.2016)

Date of mailing of the international search report

19 April 2016 (19.04.2016)

Name and mailing address of the ISA/FI  
Finnish Patent and Registration Office  
P.O. Box 1160, FI-00101 HELSINKI, Finland  
Facsimile No. +358 9 6939 5328Authorized officer  
Ari Hottinen  
Telephone No. +358 9 6939 500

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2015/050885

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                              | Relevant to claim No.           |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| X         | US 2007266028 A1 (MULLER MAX [US] et al.)<br>15 November 2007 (15.11.2007)<br>abstract; paragraphs [0012]-[0014], [0046]-[0047], [0069], [0117], [0119], [0128], [0136]; Fig. 4 | 1, 10, 19, 28,<br>37, 46, 55-58 |
| X         | US 2012191931 A1 (BRICKER ALLAN J [US] et al.)<br>26 July 2012 (26.07.2012)<br>the whole document, in particular paragraphs [0022], [0033], [0037]-[0041]; Figs. 1, 10-12       | 1, 10, 19, 28,<br>37, 46, 55-58 |
| A         | US 2008275905 A1 (HANNUKSELA MISKA [FI])<br>06 November 2008 (06.11.2008)<br>the whole document, in particular paragraphs [0004], [0007]-[0010]; Fig. 2                         | 1-58                            |
| P, X      | US 2015193494 A1 (MALAMAL VADAKITAL VINOD KUMAR [FI] et al.)<br>09 July 2015 (09.07.2015)<br>the whole document                                                                 | 1, 10, 19, 28,<br>37, 46, 55-58 |

INTERNATIONAL SEARCH REPORT

International application No.  
PCT/FI2015/050885

CLASSIFICATION OF SUBJECT MATTER

IPC  
**G06F 11/08** (2006.01)  
**H04L 29/06** (2006.01)  
**H04N 21/236** (2011.01)  
**H04N 21/4402** (2011.01)

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s)                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Publication<br>date                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| US 2002162076 A1                          | 31/10/2002          | US 7062704 B2<br>WO 02088960 A2                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 13/06/2006<br>07/11/2002                                                                                                                                                                                                                                                                                                                                   |
| US 2002023249 A1                          | 21/02/2002          | US 6954893 B2<br>AU 8650001 A<br>US 2002191258 A1<br>US 7215887 B2<br>US 2002039882 A1<br>US 7280823 B2<br>US 2003020993 A1<br>US 7386238 B2<br>US 2007223930 A1<br>US 7480462 B2<br>US 2002040388 A1<br>WO 0215438 A1                                                                                                                                                                                                                                                                         | 11/10/2005<br>25/02/2002<br>19/12/2002<br>08/05/2007<br>04/04/2002<br>09/10/2007<br>30/01/2003<br>10/06/2008<br>27/09/2007<br>20/01/2009<br>04/04/2002<br>21/02/2002                                                                                                                                                                                       |
| US 2003188231 A1                          | 02/10/2003          | US 6880149 B2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 12/04/2005                                                                                                                                                                                                                                                                                                                                                 |
| US 2007266028 A1                          | 15/11/2007          | US 8015237 B2<br>AT 426851 T<br>AT 445194 T<br>AT 463117 T<br>AU 2004234708 A1<br>AU 2004234708 B2<br>AU 2005239426 A1<br>AU 2005239426 B2<br>AU 2005323229 A1<br>AU 2005323229 A2<br>AU 2005323229 B2<br>AU 2006335156 A1<br>AU 2006335156 B2<br>AU 2006335156 B8<br>AU 2006335157 A1<br>AU 2006335157 B2<br>AU 2007100723 B4<br>AU 2007100724 B4<br>AU 2007100727 B4<br>AU 2007240187 A1<br>AU 2007240187 A2<br>AU 2007240187 B2<br>AU 2007261116 A1<br>AU 2007261116 B2<br>AU 2007300022 A1 | 06/09/2011<br>15/04/2009<br>15/10/2009<br>15/04/2010<br>11/11/2004<br>20/09/2007<br>10/11/2005<br>15/07/2010<br>13/07/2006<br>13/07/2006<br>23/06/2011<br>19/07/2007<br>02/09/2010<br>14/10/2010<br>19/07/2007<br>30/09/2010<br>27/09/2007<br>27/09/2007<br>27/09/2007<br>03/01/2008<br>17/07/2008<br>28/01/2010<br>27/12/2007<br>12/05/2011<br>03/04/2008 |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | AU 2007300022 B2         | 27/05/2010          |
|                                           |                     | AU 2007336816 A1         | 03/07/2008          |
|                                           |                     | AU 2007336816 B2         | 26/05/2011          |
|                                           |                     | AU 2007336816 C1         | 06/10/2011          |
|                                           |                     | AU 2007336832 A1         | 03/07/2008          |
|                                           |                     | AU 2007336832 B2         | 24/05/2012          |
|                                           |                     | AU 2008100083 B4         | 13/03/2008          |
|                                           |                     | AU 2008100084 B4         | 13/03/2008          |
|                                           |                     | AU 2008100376 B4         | 12/06/2008          |
|                                           |                     | AU 2008100378 B4         | 12/06/2008          |
|                                           |                     | AU 2008100379 B4         | 04/09/2008          |
|                                           |                     | AU 2008101148 B4         | 20/08/2009          |
|                                           |                     | AU 2008101150 B4         | 23/07/2009          |
|                                           |                     | AU 2008207374 A1         | 11/09/2008          |
|                                           |                     | AU 2008207374 B2         | 20/11/2008          |
|                                           |                     | AU 2008216966 A1         | 09/10/2008          |
|                                           |                     | AU 2008216966 B2         | 08/01/2009          |
|                                           |                     | AU 2008216994 A1         | 09/10/2008          |
|                                           |                     | AU 2008216994 B2         | 08/01/2009          |
|                                           |                     | AU 2009100547 B4         | 18/03/2010          |
|                                           |                     | AU 2009100658 B4         | 01/10/2009          |
|                                           |                     | AU 2009100749 B4         | 01/04/2010          |
|                                           |                     | AU 2009100750 B4         | 01/04/2010          |
|                                           |                     | AU 2010101386 B4         | 10/02/2011          |
|                                           |                     | AU 2010101387 B4         | 10/02/2011          |
|                                           |                     | AU 2010201148 A1         | 15/04/2010          |
|                                           |                     | AU 2010201148 B2         | 31/05/2012          |
|                                           |                     | AU 2011200184 A1         | 03/02/2011          |
|                                           |                     | AU 2011200184 B2         | 11/07/2013          |
|                                           |                     | CA 2464102 A1            | 01/05/2003          |
|                                           |                     | CA 2517817 A1            | 11/11/2004          |
|                                           |                     | CA 2517817 C             | 28/09/2010          |
|                                           |                     | CA 2564735 A1            | 10/11/2005          |
|                                           |                     | CA 2591164 A1            | 13/07/2006          |
|                                           |                     | CA 2707756 A1            | 11/11/2004          |
|                                           |                     | CN 1765059 A             | 26/04/2006          |
|                                           |                     | CN 1809796 A             | 26/07/2006          |
|                                           |                     | CN 1965327 A             | 16/05/2007          |
|                                           |                     | CN 1989477 A             | 27/06/2007          |
|                                           |                     | CN 100573417 C           | 23/12/2009          |
|                                           |                     | CN 101048775 A           | 03/10/2007          |
|                                           |                     | CN 101049019 A           | 03/10/2007          |
|                                           |                     | CN 101049019 B           | 03/11/2010          |
|                                           |                     | CN 101065945 A           | 31/10/2007          |
|                                           |                     | CN 101065945 B           | 10/04/2013          |
|                                           |                     | CN 101099157 A           | 02/01/2008          |
|                                           |                     | CN 101099157 B           | 11/01/2012          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | CN 101120413 A           | 06/02/2008          |
|                                           |                     | CN 101120413 B           | 07/05/2014          |
|                                           |                     | CN 101164034 A           | 16/04/2008          |
|                                           |                     | CN 101164034 B           | 13/07/2011          |
|                                           |                     | CN 101203853 A           | 18/06/2008          |
|                                           |                     | CN 101203853 B           | 04/07/2012          |
|                                           |                     | CN 101268460 A           | 17/09/2008          |
|                                           |                     | CN 101268460 B           | 03/10/2012          |
|                                           |                     | CN 101320872 A           | 10/12/2008          |
|                                           |                     | CN 101320872 B           | 04/07/2012          |
|                                           |                     | CN 101320985 A           | 10/12/2008          |
|                                           |                     | CN 101320985 B           | 02/01/2013          |
|                                           |                     | CN 101320986 A           | 10/12/2008          |
|                                           |                     | CN 101320986 B           | 30/01/2013          |
|                                           |                     | CN 101320987 A           | 10/12/2008          |
|                                           |                     | CN 101320987 B           | 02/01/2013          |
|                                           |                     | CN 101385304 A           | 11/03/2009          |
|                                           |                     | CN 101385304 B           | 07/11/2012          |
|                                           |                     | CN 101398850 A           | 01/04/2009          |
|                                           |                     | CN 101398850 B           | 31/12/2014          |
|                                           |                     | CN 101506838 A           | 12/08/2009          |
|                                           |                     | CN 101563869 A           | 21/10/2009          |
|                                           |                     | CN 101610121 A           | 23/12/2009          |
|                                           |                     | CN 101610121 B           | 08/08/2012          |
|                                           |                     | CN 101694595 A           | 14/04/2010          |
|                                           |                     | CN 101694595 B           | 06/03/2013          |
|                                           |                     | CN 101699458 A           | 28/04/2010          |
|                                           |                     | CN 101699458 B           | 29/05/2013          |
|                                           |                     | CN 101699505 A           | 28/04/2010          |
|                                           |                     | CN 101699505 B           | 17/02/2016          |
|                                           |                     | CN 102213977 A           | 12/10/2011          |
|                                           |                     | CN 102213977 B           | 26/12/2012          |
|                                           |                     | CN 102843202 A           | 26/12/2012          |
|                                           |                     | CN 102857511 A           | 02/01/2013          |
|                                           |                     | CN 102880627 A           | 16/01/2013          |
|                                           |                     | CN 102880627 B           | 02/03/2016          |
|                                           |                     | CN 102982058 A           | 20/03/2013          |
|                                           |                     | CN 102999309 A           | 27/03/2013          |
|                                           |                     | CN 103076967 A           | 01/05/2013          |
|                                           |                     | CN 103116460 A           | 22/05/2013          |
|                                           |                     | CN 103124298 A           | 29/05/2013          |
|                                           |                     | CN 103229156 A           | 31/07/2013          |
|                                           |                     | CN 103238146 A           | 07/08/2013          |
|                                           |                     | CN 103250115 A           | 14/08/2013          |
|                                           |                     | CN 103250128 A           | 14/08/2013          |
|                                           |                     | CN 103261994 A           | 21/08/2013          |
|                                           |                     | CN 103261995 A           | 21/08/2013          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | CN 103262010 A           | 21/08/2013          |
|                                           |                     | CN 103262025 A           | 21/08/2013          |
|                                           |                     | CN 103262057 A           | 21/08/2013          |
|                                           |                     | CN 103262057 B           | 10/02/2016          |
|                                           |                     | CN 103270481 A           | 28/08/2013          |
|                                           |                     | CN 103270505 A           | 28/08/2013          |
|                                           |                     | CN 103282851 A           | 04/09/2013          |
|                                           |                     | CN 103282875 A           | 04/09/2013          |
|                                           |                     | CN 103282894 A           | 04/09/2013          |
|                                           |                     | CN 103282955 A           | 04/09/2013          |
|                                           |                     | CN 103299360 A           | 11/09/2013          |
|                                           |                     | CN 103329060 A           | 25/09/2013          |
|                                           |                     | CN 103329061 A           | 25/09/2013          |
|                                           |                     | CN 103329062 A           | 25/09/2013          |
|                                           |                     | CN 103329094 A           | 25/09/2013          |
|                                           |                     | CN 103339600 A           | 02/10/2013          |
|                                           |                     | CN 103348311 A           | 09/10/2013          |
|                                           |                     | CN 103370689 A           | 23/10/2013          |
|                                           |                     | CN 103370690 A           | 23/10/2013          |
|                                           |                     | CN 103403647 A           | 20/11/2013          |
|                                           |                     | CN 103403651 A           | 20/11/2013          |
|                                           |                     | CN 103403658 A           | 20/11/2013          |
|                                           |                     | CN 103415834 A           | 27/11/2013          |
|                                           |                     | CN 103430132 A           | 04/12/2013          |
|                                           |                     | CN 103430137 A           | 04/12/2013          |
|                                           |                     | CN 103493010 A           | 01/01/2014          |
|                                           |                     | DE 202004021334 U1       | 06/09/2007          |
|                                           |                     | DE 202004021385 U1       | 07/02/2008          |
|                                           |                     | DE 202004021386 U1       | 07/02/2008          |
|                                           |                     | DE 202004021490 U1       | 21/05/2008          |
|                                           |                     | DE 202004021494 U1       | 05/06/2008          |
|                                           |                     | DE 202004021620 U1       | 27/08/2009          |
|                                           |                     | DE 202004021812 U1       | 14/04/2011          |
|                                           |                     | DE 202004021813 U1       | 31/03/2011          |
|                                           |                     | DE 202005021674 U1       | 12/03/2009          |
|                                           |                     | DE 212007000089 U1       | 27/08/2009          |
|                                           |                     | DE 602004023496 D1       | 19/11/2009          |
|                                           |                     | DE 602005020336 D1       | 12/05/2010          |
|                                           |                     | DE 602006005919 D1       | 07/05/2009          |
|                                           |                     | EP 1440402 A1            | 28/07/2004          |
|                                           |                     | EP 1471476 A1            | 27/10/2004          |
|                                           |                     | EP 1522076 A1            | 13/04/2005          |
|                                           |                     | EP 1618453 A1            | 25/01/2006          |
|                                           |                     | EP 1618453 B1            | 07/10/2009          |
|                                           |                     | EP 1618537 A1            | 25/01/2006          |
|                                           |                     | EP 1618675 A1            | 25/01/2006          |
|                                           |                     | EP 1618675 B1            | 24/04/2013          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | EP 1639440 A2            | 29/03/2006          |
|                                           |                     | EP 1745426 A2            | 24/01/2007          |
|                                           |                     | EP 1756826 A2            | 28/02/2007          |
|                                           |                     | EP 1769313 A2            | 04/04/2007          |
|                                           |                     | EP 1769313 B1            | 03/02/2016          |
|                                           |                     | EP 1805668 A2            | 11/07/2007          |
|                                           |                     | EP 1805964 A1            | 11/07/2007          |
|                                           |                     | EP 1805964 B1            | 31/03/2010          |
|                                           |                     | EP 1805991 A2            | 11/07/2007          |
|                                           |                     | EP 1834273 A1            | 19/09/2007          |
|                                           |                     | EP 1834273 B1            | 24/07/2013          |
|                                           |                     | EP 1842201 A2            | 10/10/2007          |
|                                           |                     | EP 1842201 B1            | 13/06/2012          |
|                                           |                     | EP 1880264 A1            | 23/01/2008          |
|                                           |                     | EP 1880264 B1            | 25/03/2009          |
|                                           |                     | EP 1883031 A2            | 30/01/2008          |
|                                           |                     | EP 1889183 A2            | 20/02/2008          |
|                                           |                     | EP 1889184 A2            | 20/02/2008          |
|                                           |                     | EP 1977581 A1            | 08/10/2008          |
|                                           |                     | EP 1977584 A2            | 08/10/2008          |
|                                           |                     | EP 1982504 A2            | 22/10/2008          |
|                                           |                     | EP 2019351 A2            | 28/01/2009          |
|                                           |                     | EP 2026546 A2            | 18/02/2009          |
|                                           |                     | EP 2026546 B1            | 28/11/2012          |
|                                           |                     | EP 2030160 A2            | 04/03/2009          |
|                                           |                     | EP 2034616 A1            | 11/03/2009          |
|                                           |                     | EP 2034616 B1            | 24/10/2012          |
|                                           |                     | EP 2082368 A1            | 29/07/2009          |
|                                           |                     | EP 2090000 A2            | 19/08/2009          |
|                                           |                     | EP 2090002 A2            | 19/08/2009          |
|                                           |                     | EP 2116948 A2            | 11/11/2009          |
|                                           |                     | EP 2117143 A2            | 11/11/2009          |
|                                           |                     | EP 2122838 A2            | 25/11/2009          |
|                                           |                     | EP 2209059 A1            | 21/07/2010          |
|                                           |                     | EP 2244154 A2            | 27/10/2010          |
|                                           |                     | EP 2244154 B1            | 04/06/2014          |
|                                           |                     | EP 2251763 A2            | 17/11/2010          |
|                                           |                     | EP 2251763 B1            | 26/02/2014          |
|                                           |                     | EP 2254043 A1            | 24/11/2010          |
|                                           |                     | EP 2256740 A2            | 01/12/2010          |
|                                           |                     | EP 2256740 B1            | 09/09/2015          |
|                                           |                     | EP 2256741 A2            | 01/12/2010          |
|                                           |                     | EP 2264712 A2            | 22/12/2010          |
|                                           |                     | EP 2270730 A1            | 05/01/2011          |
|                                           |                     | EP 2273787 A2            | 12/01/2011          |
|                                           |                     | EP 2285108 A2            | 16/02/2011          |
|                                           |                     | EP 2302528 A2            | 30/03/2011          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | EP 2312847 A2            | 20/04/2011          |
|                                           |                     | EP 2328320 A2            | 01/06/2011          |
|                                           |                     | EP 2330792 A1            | 08/06/2011          |
|                                           |                     | EP 2333777 A2            | 15/06/2011          |
|                                           |                     | EP 2357623 A1            | 17/08/2011          |
|                                           |                     | EP 2363777 A1            | 07/09/2011          |
|                                           |                     | EP 2565751 A1            | 06/03/2013          |
|                                           |                     | EP 2622432 A2            | 07/08/2013          |
|                                           |                     | EP 2622433 A2            | 07/08/2013          |
|                                           |                     | EP 2622434 A2            | 07/08/2013          |
|                                           |                     | EP 2622436 A1            | 07/08/2013          |
|                                           |                     | EP 2622438 A2            | 07/08/2013          |
|                                           |                     | EP 2622439 A2            | 07/08/2013          |
|                                           |                     | EP 2622443 A1            | 07/08/2013          |
|                                           |                     | EP 2622444 A2            | 07/08/2013          |
|                                           |                     | EP 2622446 A2            | 07/08/2013          |
|                                           |                     | EP 2622447 A2            | 07/08/2013          |
|                                           |                     | EP 2622448 A2            | 07/08/2013          |
|                                           |                     | EP 2622449 A1            | 07/08/2013          |
|                                           |                     | EP 2622450 A1            | 07/08/2013          |
|                                           |                     | EP 2622455 A2            | 07/08/2013          |
|                                           |                     | EP 2622462 A2            | 07/08/2013          |
|                                           |                     | EP 2622463 A2            | 07/08/2013          |
|                                           |                     | EP 2622464 A2            | 07/08/2013          |
|                                           |                     | EP 2622488 A2            | 07/08/2013          |
|                                           |                     | EP 2622490 A2            | 07/08/2013          |
|                                           |                     | EP 2622491 A2            | 07/08/2013          |
|                                           |                     | EP 2622494 A2            | 07/08/2013          |
|                                           |                     | EP 2622596 A1            | 07/08/2013          |
|                                           |                     | EP 2622597 A2            | 07/08/2013          |
|                                           |                     | EP 2641226 A2            | 25/09/2013          |
|                                           |                     | ES 2324999 T3            | 21/08/2009          |
|                                           |                     | ES 2422307 T3            | 10/09/2013          |
|                                           |                     | GB 0314394 D0            | 23/07/2003          |
|                                           |                     | GB 2387001 A             | 01/10/2003          |
|                                           |                     | GB 2387001 B             | 02/02/2005          |
|                                           |                     | GB 0425738 D0            | 22/12/2004          |
|                                           |                     | GB 2405718 A             | 09/03/2005          |
|                                           |                     | GB 2405718 B             | 29/03/2006          |
|                                           |                     | GB 0425740 D0            | 22/12/2004          |
|                                           |                     | GB 2405719 A             | 09/03/2005          |
|                                           |                     | GB 2405719 B             | 29/03/2006          |
|                                           |                     | GB 0425742 D0            | 22/12/2004          |
|                                           |                     | GB 2405720 A             | 09/03/2005          |
|                                           |                     | GB 2405720 B             | 29/03/2006          |
|                                           |                     | GB 0813588 D0            | 03/09/2008          |
|                                           |                     | GB 2449783 A             | 03/12/2008          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | GB 2449783 B             | 15/06/2011          |
|                                           |                     | GB 0813592 D0            | 03/09/2008          |
|                                           |                     | GB 2447598 A             | 17/09/2008          |
|                                           |                     | GB 2447598 B             | 22/12/2010          |
|                                           |                     | GB 0906583 D0            | 27/05/2009          |
|                                           |                     | GB 2456095 A             | 08/07/2009          |
|                                           |                     | GB 2456095 B             | 27/07/2011          |
|                                           |                     | GB 0909123 D0            | 01/07/2009          |
|                                           |                     | GB 2456473 A             | 22/07/2009          |
|                                           |                     | GB 2456473 B             | 17/08/2011          |
|                                           |                     | GB 0909613 D0            | 15/07/2009          |
|                                           |                     | GB 2456731 A             | 29/07/2009          |
|                                           |                     | GB 2456731 B             | 16/02/2011          |
|                                           |                     | GB 0911578 D0            | 12/08/2009          |
|                                           |                     | GB 2459211 A             | 21/10/2009          |
|                                           |                     | GB 2459211 B             | 16/02/2011          |
|                                           |                     | HK 1057631 A1            | 23/09/2005          |
|                                           |                     | HK 1072821 A1            | 14/07/2006          |
|                                           |                     | HK 1072822 A1            | 16/06/2006          |
|                                           |                     | HK 1072823 A1            | 16/06/2006          |
|                                           |                     | HK 1080187 A1            | 24/12/2009          |
|                                           |                     | HK 1080230 A1            | 09/08/2013          |
|                                           |                     | HK 1106895 A1            | 23/07/2010          |
|                                           |                     | HK 1109951 A1            | 05/10/2012          |
|                                           |                     | HK 1112769 A1            | 06/11/2009          |
|                                           |                     | HK 1114196 A1            | 21/09/2012          |
|                                           |                     | HK 1121829 A1            | 05/04/2013          |
|                                           |                     | HK 1126576 A1            | 12/04/2013          |
|                                           |                     | HK 1126580 A1            | 03/05/2013          |
|                                           |                     | HK 1126582 A1            | 01/11/2013          |
|                                           |                     | HK 1126903 A1            | 11/10/2013          |
|                                           |                     | HK 1129168 A1            | 08/02/2013          |
|                                           |                     | HK 1129180 A1            | 15/03/2013          |
|                                           |                     | HK 1133494 A1            | 23/12/2011          |
|                                           |                     | HK 1150083 A1            | 15/08/2014          |
|                                           |                     | HK 1151126 A1            | 22/01/2016          |
|                                           |                     | IN 659KON2015 A          | 17/07/2015          |
|                                           |                     | JP 2005507130 A          | 10/03/2005          |
|                                           |                     | JP 4204977 B2            | 07/01/2009          |
|                                           |                     | JP 2008518330 A          | 29/05/2008          |
|                                           |                     | JP 4388122 B2            | 24/12/2009          |
|                                           |                     | JP 2009110645 A          | 21/05/2009          |
|                                           |                     | JP 4406466 B2            | 27/01/2010          |
|                                           |                     | JP 2008527444 A          | 24/07/2008          |
|                                           |                     | JP 4571677 B2            | 27/10/2010          |
|                                           |                     | JP 2008507062 A          | 06/03/2008          |
|                                           |                     | JP 4664362 B2            | 06/04/2011          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | JP 2010282633 A          | 16/12/2010          |
|                                           |                     | JP 4695220 B2            | 08/06/2011          |
|                                           |                     | JP 2008518354 A          | 29/05/2008          |
|                                           |                     | JP 4695653 B2            | 08/06/2011          |
|                                           |                     | JP 2007535077 A          | 29/11/2007          |
|                                           |                     | JP 4739328 B2            | 03/08/2011          |
|                                           |                     | JP 2008527537 A          | 24/07/2008          |
|                                           |                     | JP 4787271 B2            | 05/10/2011          |
|                                           |                     | JP 2006524874 A          | 02/11/2006          |
|                                           |                     | JP 4789802 B2            | 12/10/2011          |
|                                           |                     | JP 2006524972 A          | 02/11/2006          |
|                                           |                     | JP 4921966 B2            | 25/04/2012          |
|                                           |                     | JP 2008541298 A          | 20/11/2008          |
|                                           |                     | JP 4995815 B2            | 08/08/2012          |
|                                           |                     | JP 2008546057 A          | 18/12/2008          |
|                                           |                     | JP 5038295 B2            | 03/10/2012          |
|                                           |                     | JP 2009005367 A          | 08/01/2009          |
|                                           |                     | JP 5066491 B2            | 07/11/2012          |
|                                           |                     | JP 2011054188 A          | 17/03/2011          |
|                                           |                     | JP 5244884 B2            | 24/07/2013          |
|                                           |                     | JP 2010262733 A          | 18/11/2010          |
|                                           |                     | JP 5361819 B2            | 04/12/2013          |
|                                           |                     | JP 2008020910 A          | 31/01/2008          |
|                                           |                     | JP 5528661 B2            | 25/06/2014          |
|                                           |                     | JP 2012150832 A          | 09/08/2012          |
|                                           |                     | JP 5586647 B2            | 10/09/2014          |
|                                           |                     | JP 2013543620 A          | 05/12/2013          |
|                                           |                     | JP 5675997 B2            | 25/02/2015          |
|                                           |                     | JP 2012075152 A          | 12/04/2012          |
|                                           |                     | JP 5679951 B2            | 04/03/2015          |
|                                           |                     | JP 2014053966 A          | 20/03/2014          |
|                                           |                     | JP 5786013 B2            | 30/09/2015          |
|                                           |                     | JP 2013542516 A          | 21/11/2013          |
|                                           |                     | JP 5827337 B2            | 02/12/2015          |
|                                           |                     | JP 2013540316 A          | 31/10/2013          |
|                                           |                     | JP 5847826 B2            | 27/01/2016          |
|                                           |                     | JP 2005533333 A          | 04/11/2005          |
|                                           |                     | JP 2006524875 A          | 02/11/2006          |
|                                           |                     | JP 2006524877 A          | 02/11/2006          |
|                                           |                     | JP 2010183571 A          | 19/08/2010          |
|                                           |                     | JP 2010186493 A          | 26/08/2010          |
|                                           |                     | JP 2013012290 A          | 17/01/2013          |
|                                           |                     | JP 2013539137 A          | 17/10/2013          |
|                                           |                     | JP 2013540318 A          | 31/10/2013          |
|                                           |                     | JP 2013541100 A          | 07/11/2013          |
|                                           |                     | JP 2013542512 A          | 21/11/2013          |
|                                           |                     | JP 2013542515 A          | 21/11/2013          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | JP 2013543618 A          | 05/12/2013          |
|                                           |                     | JP 2013543621 A          | 05/12/2013          |
|                                           |                     | JP 2013544410 A          | 12/12/2013          |
|                                           |                     | JP 2013545168 A          | 19/12/2013          |
|                                           |                     | JP 2013546043 A          | 26/12/2013          |
|                                           |                     | JP 2013546044 A          | 26/12/2013          |
|                                           |                     | JP 2013546045 A          | 26/12/2013          |
|                                           |                     | JP 2013546046 A          | 26/12/2013          |
|                                           |                     | JP 2013546047 A          | 26/12/2013          |
|                                           |                     | JP 2013546048 A          | 26/12/2013          |
|                                           |                     | JP 2013546049 A          | 26/12/2013          |
|                                           |                     | JP 2013546050 A          | 26/12/2013          |
|                                           |                     | JP 2014500992 A          | 16/01/2014          |
|                                           |                     | JP 2014508977 A          | 10/04/2014          |
|                                           |                     | KR 20060006050 A         | 18/01/2006          |
|                                           |                     | KR 100688414 B1          | 02/03/2007          |
|                                           |                     | KR 20040058213 A         | 03/07/2004          |
|                                           |                     | KR 100718613 B1          | 16/05/2007          |
|                                           |                     | KR 20060052670 A         | 19/05/2006          |
|                                           |                     | KR 100732590 B1          | 27/06/2007          |
|                                           |                     | KR 20070044489 A         | 27/04/2007          |
|                                           |                     | KR 100869630 B1          | 21/11/2008          |
|                                           |                     | KR 20070031460 A         | 19/03/2007          |
|                                           |                     | KR 100947697 B1          | 16/03/2010          |
|                                           |                     | KR 20080090577 A         | 08/10/2008          |
|                                           |                     | KR 100957908 B1          | 13/05/2010          |
|                                           |                     | KR 20080021846 A         | 07/03/2008          |
|                                           |                     | KR 100988404 B1          | 18/10/2010          |
|                                           |                     | KR 20060004923 A         | 16/01/2006          |
|                                           |                     | KR 100996317 B1          | 23/11/2010          |
|                                           |                     | KR 20080090578 A         | 08/10/2008          |
|                                           |                     | KR 100996365 B1          | 23/11/2010          |
|                                           |                     | KR 20090046961 A         | 11/05/2009          |
|                                           |                     | KR 100997309 B1          | 29/11/2010          |
|                                           |                     | KR 20070046170 A         | 02/05/2007          |
|                                           |                     | KR 101001048 B1          | 14/12/2010          |
|                                           |                     | KR 20090132647 A         | 30/12/2009          |
|                                           |                     | KR 101037838 B1          | 31/05/2011          |
|                                           |                     | KR 20080081367 A         | 09/09/2008          |
|                                           |                     | KR 101043536 B1          | 23/06/2011          |
|                                           |                     | KR 20090117963 A         | 16/11/2009          |
|                                           |                     | KR 101143704 B1          | 09/05/2012          |
|                                           |                     | KR 20120043000 A         | 03/05/2012          |
|                                           |                     | KR 101380433 B1          | 01/04/2014          |
|                                           |                     | KR 20070046171 A         | 02/05/2007          |
|                                           |                     | KR 20070104393 A         | 25/10/2007          |
|                                           |                     | KR 20090027777 A         | 17/03/2009          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | KR 20100093142 A         | 24/08/2010          |
|                                           |                     | KR 20120102783 A         | 18/09/2012          |
|                                           |                     | MX 2013003176 A          | 01/08/2013          |
|                                           |                     | MX 2013003177 A          | 01/08/2013          |
|                                           |                     | MX 2013003247 A          | 30/10/2013          |
|                                           |                     | MX 2013003249 A          | 28/10/2013          |
|                                           |                     | MX 2013003424 A          | 28/10/2013          |
|                                           |                     | MX 2013003427 A          | 28/10/2013          |
|                                           |                     | MX 2013003515 A          | 26/09/2013          |
|                                           |                     | MX 2013003517 A          | 26/09/2013          |
|                                           |                     | TW 200843394 A           | 01/11/2008          |
|                                           |                     | TW I362852 B             | 21/04/2012          |
|                                           |                     | TW 200845639 A           | 16/11/2008          |
|                                           |                     | TW I441471 B             | 11/06/2014          |
|                                           |                     | US D521936 S1            | 30/05/2006          |
|                                           |                     | US D579876 S1            | 04/11/2008          |
|                                           |                     | US D601097 S1            | 29/09/2009          |
|                                           |                     | US 7433546 B2            | 07/10/2008          |
|                                           |                     | US 7478323 B2            | 13/01/2009          |
|                                           |                     | US 7525216 B2            | 28/04/2009          |
|                                           |                     | US 7536565 B2            | 19/05/2009          |
|                                           |                     | US 7565036 B2            | 21/07/2009          |
|                                           |                     | US 2006015757 A1         | 19/01/2006          |
|                                           |                     | US 7581119 B2            | 25/08/2009          |
|                                           |                     | US 7593782 B2            | 22/09/2009          |
|                                           |                     | US 7607019 B2            | 20/10/2009          |
|                                           |                     | US 7623740 B2            | 24/11/2009          |
|                                           |                     | US 2004224638 A1         | 11/11/2004          |
|                                           |                     | US 7627343 B2            | 01/12/2009          |
|                                           |                     | US 7650507 B2            | 19/01/2010          |
|                                           |                     | US 7680849 B2            | 16/03/2010          |
|                                           |                     | US 7686215 B2            | 30/03/2010          |
|                                           |                     | US 7706637 B2            | 27/04/2010          |
|                                           |                     | US 7719830 B2            | 18/05/2010          |
|                                           |                     | US 7724716 B2            | 25/05/2010          |
|                                           |                     | US 7751853 B2            | 06/07/2010          |
|                                           |                     | US 2003079038 A1         | 24/04/2003          |
|                                           |                     | US 7765326 B2            | 27/07/2010          |
|                                           |                     | US 7769903 B2            | 03/08/2010          |
|                                           |                     | US 7783070 B2            | 24/08/2010          |
|                                           |                     | US 2005021478 A1         | 27/01/2005          |
|                                           |                     | US 7797242 B2            | 14/09/2010          |
|                                           |                     | US 7797446 B2            | 14/09/2010          |
|                                           |                     | US 7816811 B2            | 19/10/2010          |
|                                           |                     | US 7823214 B2            | 26/10/2010          |
|                                           |                     | US 7827162 B2            | 02/11/2010          |
|                                           |                     | US 2005240661 A1         | 27/10/2005          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | US 7827259 B2            | 02/11/2010          |
|                                           |                     | US 7831199 B2            | 09/11/2010          |
|                                           |                     | US 7844498 B2            | 30/11/2010          |
|                                           |                     | US 7844548 B2            | 30/11/2010          |
|                                           |                     | US 2005071780 A1         | 31/03/2005          |
|                                           |                     | US 7853893 B2            | 14/12/2010          |
|                                           |                     | US 7856564 B2            | 21/12/2010          |
|                                           |                     | US 2005278377 A1         | 15/12/2005          |
|                                           |                     | US 7860830 B2            | 28/12/2010          |
|                                           |                     | US 7865745 B2            | 04/01/2011          |
|                                           |                     | US 7881564 B2            | 01/02/2011          |
|                                           |                     | US 7889497 B2            | 15/02/2011          |
|                                           |                     | US 7890783 B2            | 15/02/2011          |
|                                           |                     | US 2005050345 A1         | 03/03/2005          |
|                                           |                     | US 7895661 B2            | 22/02/2011          |
|                                           |                     | US 7899714 B2            | 01/03/2011          |
|                                           |                     | US 7933117 B2            | 26/04/2011          |
|                                           |                     | US 7958441 B2            | 07/06/2011          |
|                                           |                     | US 7962634 B2            | 14/06/2011          |
|                                           |                     | US 7966362 B2            | 21/06/2011          |
|                                           |                     | US 8020762 B2            | 20/09/2011          |
|                                           |                     | US 8050714 B2            | 01/11/2011          |
|                                           |                     | US 8072956 B2            | 06/12/2011          |
|                                           |                     | US 8078224 B2            | 13/12/2011          |
|                                           |                     | US 8103793 B2            | 24/01/2012          |
|                                           |                     | US 8150937 B2            | 03/04/2012          |
|                                           |                     | US 8151259 B2            | 03/04/2012          |
|                                           |                     | US 2004268451 A1         | 30/12/2004          |
|                                           |                     | US 8161411 B2            | 17/04/2012          |
|                                           |                     | US 8161567 B2            | 17/04/2012          |
|                                           |                     | US 8165634 B2            | 24/04/2012          |
|                                           |                     | US 8180895 B2            | 15/05/2012          |
|                                           |                     | US 8190205 B2            | 29/05/2012          |
|                                           |                     | US 8200629 B2            | 12/06/2012          |
|                                           |                     | US 8238971 B2            | 07/08/2012          |
|                                           |                     | US 8245924 B2            | 21/08/2012          |
|                                           |                     | US 8259444 B2            | 04/09/2012          |
|                                           |                     | US 8271038 B2            | 18/09/2012          |
|                                           |                     | US 2005193094 A1         | 01/09/2005          |
|                                           |                     | US 8291320 B2            | 16/10/2012          |
|                                           |                     | US 8332668 B2            | 11/12/2012          |
|                                           |                     | US 8359348 B2            | 22/01/2013          |
|                                           |                     | US 8370419 B2            | 05/02/2013          |
|                                           |                     | US 8412763 B2            | 02/04/2013          |
|                                           |                     | US 8458184 B2            | 04/06/2013          |
|                                           |                     | US 8467829 B2            | 18/06/2013          |
|                                           |                     | US 8473479 B2            | 25/06/2013          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | US 8489468 B2            | 16/07/2013          |
|                                           |                     | US 8495246 B2            | 23/07/2013          |
|                                           |                     | US 8504936 B2            | 06/08/2013          |
|                                           |                     | US 8516035 B2            | 20/08/2013          |
|                                           |                     | US 8527892 B2            | 03/09/2013          |
|                                           |                     | US 8581449 B2            | 12/11/2013          |
|                                           |                     | US 8588860 B2            | 19/11/2013          |
|                                           |                     | US 8599106 B2            | 03/12/2013          |
|                                           |                     | US 8626952 B2            | 07/01/2014          |
|                                           |                     | US 8646687 B2            | 11/02/2014          |
|                                           |                     | US 8648825 B2            | 11/02/2014          |
|                                           |                     | US 8659565 B2            | 25/02/2014          |
|                                           |                     | US 8665215 B2            | 04/03/2014          |
|                                           |                     | US 8683009 B2            | 25/03/2014          |
|                                           |                     | US 8683496 B2            | 25/03/2014          |
|                                           |                     | US 8694024 B2            | 08/04/2014          |
|                                           |                     | US 8698751 B2            | 15/04/2014          |
|                                           |                     | US 8706159 B2            | 22/04/2014          |
|                                           |                     | US 8726294 B2            | 13/05/2014          |
|                                           |                     | US 8732373 B2            | 20/05/2014          |
|                                           |                     | US 8749484 B2            | 10/06/2014          |
|                                           |                     | US 8763079 B2            | 24/06/2014          |
|                                           |                     | US 8773378 B2            | 08/07/2014          |
|                                           |                     | US 8788423 B2            | 22/07/2014          |
|                                           |                     | US 8793608 B2            | 29/07/2014          |
|                                           |                     | US 8812051 B2            | 19/08/2014          |
|                                           |                     | US 8819180 B2            | 26/08/2014          |
|                                           |                     | US 8819705 B2            | 26/08/2014          |
|                                           |                     | US 8832577 B2            | 09/09/2014          |
|                                           |                     | US 8838095 B2            | 16/09/2014          |
|                                           |                     | US 8842057 B2            | 23/09/2014          |
|                                           |                     | US 8842080 B2            | 23/09/2014          |
|                                           |                     | US 8866748 B2            | 21/10/2014          |
|                                           |                     | US 8866763 B2            | 21/10/2014          |
|                                           |                     | US 8866764 B2            | 21/10/2014          |
|                                           |                     | US 8868135 B2            | 21/10/2014          |
|                                           |                     | US 8872727 B2            | 28/10/2014          |
|                                           |                     | US 8872731 B2            | 28/10/2014          |
|                                           |                     | US 8874894 B2            | 28/10/2014          |
|                                           |                     | US 8875050 B2            | 28/10/2014          |
|                                           |                     | US 8878794 B2            | 04/11/2014          |
|                                           |                     | US 8880712 B2            | 04/11/2014          |
|                                           |                     | US 8881053 B2            | 04/11/2014          |
|                                           |                     | US 8884841 B2            | 11/11/2014          |
|                                           |                     | US 8890768 B2            | 18/11/2014          |
|                                           |                     | US 8898443 B2            | 25/11/2014          |
|                                           |                     | US 8903377 B2            | 02/12/2014          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | US 8904165 B2            | 02/12/2014          |
|                                           |                     | US 8907904 B2            | 09/12/2014          |
|                                           |                     | US 8907906 B2            | 09/12/2014          |
|                                           |                     | US 8910061 B2            | 09/12/2014          |
|                                           |                     | US 8917221 B2            | 23/12/2014          |
|                                           |                     | US 8930846 B2            | 06/01/2015          |
|                                           |                     | US 8933949 B2            | 13/01/2015          |
|                                           |                     | US 8938217 B2            | 20/01/2015          |
|                                           |                     | US 8943434 B2            | 27/01/2015          |
|                                           |                     | US 8947376 B2            | 03/02/2015          |
|                                           |                     | US 8957905 B2            | 17/02/2015          |
|                                           |                     | US 8959445 B2            | 17/02/2015          |
|                                           |                     | US 8963840 B2            | 24/02/2015          |
|                                           |                     | US 8963853 B2            | 24/02/2015          |
|                                           |                     | US 8963939 B2            | 24/02/2015          |
|                                           |                     | US 8966379 B2            | 24/02/2015          |
|                                           |                     | US 8966470 B2            | 24/02/2015          |
|                                           |                     | US 8984440 B2            | 17/03/2015          |
|                                           |                     | US 8990712 B2            | 24/03/2015          |
|                                           |                     | US 8990713 B2            | 24/03/2015          |
|                                           |                     | US 8993866 B2            | 31/03/2015          |
|                                           |                     | US 8994671 B2            | 31/03/2015          |
|                                           |                     | US 8994713 B2            | 31/03/2015          |
|                                           |                     | US 9001103 B2            | 07/04/2015          |
|                                           |                     | US 9001158 B2            | 07/04/2015          |
|                                           |                     | US 9003311 B2            | 07/04/2015          |
|                                           |                     | US 9019214 B2            | 28/04/2015          |
|                                           |                     | US 9026709 B2            | 05/05/2015          |
|                                           |                     | US 9026923 B2            | 05/05/2015          |
|                                           |                     | US 9026930 B2            | 05/05/2015          |
|                                           |                     | US 9026937 B2            | 05/05/2015          |
|                                           |                     | US 9046992 B2            | 02/06/2015          |
|                                           |                     | US 9047038 B2            | 02/06/2015          |
|                                           |                     | US 9047047 B2            | 02/06/2015          |
|                                           |                     | US 9047102 B2            | 02/06/2015          |
|                                           |                     | US 9049213 B2            | 02/06/2015          |
|                                           |                     | US 9052800 B2            | 09/06/2015          |
|                                           |                     | US 9052801 B2            | 09/06/2015          |
|                                           |                     | US 9058153 B2            | 16/06/2015          |
|                                           |                     | US 9060006 B2            | 16/06/2015          |
|                                           |                     | US 9063694 B2            | 23/06/2015          |
|                                           |                     | US 9069518 B2            | 30/06/2015          |
|                                           |                     | US 9071625 B2            | 30/06/2015          |
|                                           |                     | US 9077731 B2            | 07/07/2015          |
|                                           |                     | US 9087061 B2            | 21/07/2015          |
|                                           |                     | US 9092183 B2            | 28/07/2015          |
|                                           |                     | US 9092190 B2            | 28/07/2015          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | US 9092191 B2            | 28/07/2015          |
|                                           |                     | US 9098437 B2            | 04/08/2015          |
|                                           |                     | US 9104365 B2            | 11/08/2015          |
|                                           |                     | US 9122440 B2            | 01/09/2015          |
|                                           |                     | US 9122441 B2            | 01/09/2015          |
|                                           |                     | US 9128582 B2            | 08/09/2015          |
|                                           |                     | US 9128583 B2            | 08/09/2015          |
|                                           |                     | US 9130686 B2            | 08/09/2015          |
|                                           |                     | US 9134756 B2            | 15/09/2015          |
|                                           |                     | US 9134877 B2            | 15/09/2015          |
|                                           |                     | US 9141135 B2            | 22/09/2015          |
|                                           |                     | US 9141328 B2            | 22/09/2015          |
|                                           |                     | US 9146585 B2            | 29/09/2015          |
|                                           |                     | US 9152176 B2            | 06/10/2015          |
|                                           |                     | US 9152179 B2            | 06/10/2015          |
|                                           |                     | US 9152371 B2            | 06/10/2015          |
|                                           |                     | US 9152582 B2            | 06/10/2015          |
|                                           |                     | US 9158494 B2            | 13/10/2015          |
|                                           |                     | US 9160796 B2            | 13/10/2015          |
|                                           |                     | US 9164540 B2            | 20/10/2015          |
|                                           |                     | US 9182935 B2            | 10/11/2015          |
|                                           |                     | US 9182937 B2            | 10/11/2015          |
|                                           |                     | US 9189018 B2            | 17/11/2015          |
|                                           |                     | US 9189773 B2            | 17/11/2015          |
|                                           |                     | US 9195330 B2            | 24/11/2015          |
|                                           |                     | US 9207717 B2            | 08/12/2015          |
|                                           |                     | US 9208477 B2            | 08/12/2015          |
|                                           |                     | US 9213365 B2            | 15/12/2015          |
|                                           |                     | US 9213431 B2            | 15/12/2015          |
|                                           |                     | US 9213516 B2            | 15/12/2015          |
|                                           |                     | US 9213517 B2            | 15/12/2015          |
|                                           |                     | US 9218021 B2            | 22/12/2015          |
|                                           |                     | US 9218154 B2            | 22/12/2015          |
|                                           |                     | US 9223426 B2            | 29/12/2015          |
|                                           |                     | US 9223535 B2            | 29/12/2015          |
|                                           |                     | US 9229474 B2            | 05/01/2016          |
|                                           |                     | US 9235233 B2            | 12/01/2016          |
|                                           |                     | US 9235374 B2            | 12/01/2016          |
|                                           |                     | US 9235828 B2            | 12/01/2016          |
|                                           |                     | US 9268518 B2            | 23/02/2016          |
|                                           |                     | US 9268830 B2            | 23/02/2016          |
|                                           |                     | US 9285957 B2            | 15/03/2016          |
|                                           |                     | US 9304540 B2            | 05/04/2016          |
|                                           |                     | US 2003167318 A1         | 04/09/2003          |
|                                           |                     | US 2004215534 A1         | 28/10/2004          |
|                                           |                     | US 2004216108 A1         | 28/10/2004          |
|                                           |                     | US 2004254883 A1         | 16/12/2004          |

**INTERNATIONAL SEARCH REPORT**  
**Information on Patent Family Members**

International application No.  
PCT/FI2015/050885

| Patent document<br>cited in search report | Publication<br>date | Patent family members(s) | Publication<br>date |
|-------------------------------------------|---------------------|--------------------------|---------------------|
|                                           |                     | US 2005203959 A1         | 15/09/2005          |
|                                           |                     | US 2005240494 A1         | 27/10/2005          |
|                                           |                     | US 2006015378 A1         | 19/01/2006          |
| .....                                     |                     |                          |                     |
| US 2012191931 A1                          | 26/07/2012          | US 8364917 B2            | 29/01/2013          |
|                                           |                     | US 7584338 B1            | 01/09/2009          |
|                                           |                     | US 2009292746 A1         | 26/11/2009          |
|                                           |                     | US 8028138 B2            | 27/09/2011          |
|                                           |                     | US 2011307441 A1         | 15/12/2011          |
|                                           |                     | US 8176277 B2            | 08/05/2012          |
| .....                                     |                     |                          |                     |
| US 2008275905 A1                          | 06/11/2008          | US 8078644 B2            | 13/12/2011          |
|                                           |                     | AP 200905050 D0          | 31/12/2009          |
|                                           |                     | AP 2923 A                | 31/05/2014          |
|                                           |                     | CA 2684851 A1            | 13/11/2008          |
|                                           |                     | CA 2684851 C             | 24/11/2015          |
|                                           |                     | CN 101675435 A           | 17/03/2010          |
|                                           |                     | CN 101675435 B           | 06/02/2013          |
|                                           |                     | EP 2145270 A2            | 20/01/2010          |
|                                           |                     | JP 2010526468 A          | 29/07/2010          |
|                                           |                     | JP 5130352 B2            | 30/01/2013          |
|                                           |                     | KR 20100028546 A         | 12/03/2010          |
|                                           |                     | KR 101107815 B1          | 06/02/2012          |
|                                           |                     | MX 2009011873 A          | 18/11/2009          |
|                                           |                     | RU 2434277 C2            | 20/11/2011          |
|                                           |                     | TW 200901761 A           | 01/01/2009          |
|                                           |                     | TW I500325 B             | 11/09/2015          |
|                                           |                     | WO 2008135932 A2         | 13/11/2008          |
| .....                                     |                     |                          |                     |
| US 2015193494 A1                          | 09/07/2015          | WO 2015104450 A1         | 16/07/2015          |
| .....                                     |                     |                          |                     |