

AUDIO AND VIDEO PROCESSING FOR IDENTIFICATION AND SYNCHRONISATION

FIELD OF INVENTION

[0001] Aspects of the present disclosure relate to establishing secure communication between devices. Specifically, but not exclusively, aspects of the present disclosure are directed to establishing secure communication for output synchronisation between devices using audio or video signals. Specifically, but not exclusively, aspects of the present disclosure are directed to audio or video signal processing for relevant information retrieval and/or establishing bidirectional communication.

BACKGROUND

[0002] The current media consumption landscape is characterized by a vast array of content available through various channels like television, radio, and streaming services. However, despite this abundance, viewers often encounter difficulties in identifying specific content, especially when they lack prior information like titles or actors. Traditional search methods, reliant on textual input, fall short in real-time identification and contextual understanding of the content. Current technologies primarily focus on metadata or manual search inputs for content identification. This approach is not only time-consuming but also ineffective in situations where viewers are unaware of the content's specifics. Moreover, these methods do not provide real-time synchronization with the content, limiting the depth of user interaction. Multiple search interactions between a user and user device are frequently required, resulting in poor man-machine interaction and an increase in battery consumption and required computational resources, while efficiency is greatly reduced.

[0003] Examples of current technologies include Amazon X-Ray (TM) functionality and other similar technologies, where subject information is displayed to a user based on streamed content being displayed at a specific time point. However, there is no disclosure of displaying relevant subject information based on content being broadcast from a device, application, or service not associated with the user device. At best, an external device such as a phone must be in continual communication with a display device such as a television in order to provide time-relevant information, which is inefficient, requires additional computational resources, and is limited in functionality. Another example, US9837127B2, discloses post-production insertion of audio cues into media files, with particular items marked and placed in a timeline and the marker inserted into the tagged audio track. There is no synchronization possible when audio cues are not detectable, such as when in a noisy environment or listening to a media file that does not have the markers inserted. Additionally, there is no solution to problems with identifying, extracting, and generating relevant content information associated with the media files, resulting in lengthy user-implemented searches, lengthy/inaccurate data retrieval, and generally poor interaction between a user device and media content.

[0004] Additionally, associating devices is complex. The field of secure communication and output synchronization focuses on ensuring that data exchanged between devices is protected

against unauthorized access, tampering, and eavesdropping while maintaining consistent and synchronized outputs across devices. This involves encryption protocols, authentication mechanisms, and synchronization techniques to achieve real-time or near-real-time consistency, especially in distributed systems, IoT, and collaborative environments. However, challenges persist. Traditional methods often rely on visual or manual input for device pairing and synchronization, which can be cumbersome and prone to user error. Moreover, ensuring secure communication while maintaining real-time synchronization of outputs between devices remains a significant technical hurdle. These issues are particularly pronounced in environments where multiple devices need to interact seamlessly and securely, such as in smart homes, collaborative workspaces, and multimedia streaming services.

[0005] One of the primary problems is the secure and efficient pairing of devices. Existing systems often suffer from limited security due to reliance on traditional methods like Bluetooth pairing or QR code scanning, which are vulnerable to man-in-the-middle attacks and unauthorized access. These methods are easily detectable and interceptable, which compromises the overall security of the system. Additionally, the user experience is cumbersome and error-prone, requiring manual input such as entering PINs or scanning QR codes. This complexity leads to a less intuitive and more frustrating setup process, increasing the likelihood of user errors. Another beneficial issue is the synchronization of outputs between devices. For example, in multimedia applications, ensuring that audio and video outputs are perfectly synchronized across different devices is challenging. Existing systems struggle to maintain real-time synchronization of outputs between devices. Factors like latency, network jitter, and varying processing times can cause desynchronization, particularly in multimedia applications where audio and video need to be perfectly aligned. Furthermore, current methods lack dynamic connection management, often requiring manual intervention to sustain or terminate connections, which reduces the system's reliability and robustness. The absence of comprehensive tools like Software Development Kits (SDKs) and pre-defined instructions for secure communication and synchronization also poses challenges for developers, hindering the widespread adoption and integration of advanced technologies, especially regarding older media devices. These issues, among others, highlight the need for optimization in secure communication and synchronization technologies.

SUMMARY OF INVENTION

[0006] An aspect of the present disclosure presents an innovative audio analysis system for improved content discovery and interaction, such as for television and radio broadcasts. Unlike conventional methods, the enclosed system allows users to identify and engage with media content without prior knowledge of its title or specifics. This novel technology is not only directed towards identifying the content being played, but is also directed towards synchronizing or otherwise linking content discovery and interaction with the content being played in real-time, providing increases in efficiency, accuracy, and reliability.

[0007] According to an aspect of the present disclosure, there is provided a system for media content interaction. The system is configured to process a known media content at a workstation device; interact with a media content at a user device; and transmit content information to the user device.

[0008] According to a further aspect of the present disclosure, there is provided a method for media content interaction. The method comprises processing a known media content at the workstation device, wherein processing the known media content comprises: processing a known media content at a workstation device; interacting with a media content at a user device; and transmitting content information to the user device.

[0009] Processing the known media content comprises obtaining the known media content, where the media content comprises a known audio track. A timeline from the known media content is then generated, wherein the timeline comprises a plurality of timepoints. For each timepoint of the timeline, an audio tag comprising the identification data is embedded within the known audio track at frequencies detectable by the microphone and less detectable to a user of the user device. The audio track with embedded audio tags is an embedded audio track. Content information associated with the identification data is generated by collating information associated with one or more subject extracted from the known media content at each timepoint. Both the embedded audio track and the content information associated with each timepoint of the timeline is then stored.

[0010] Interacting with the media content comprises monitoring a portion of an unknown audio track using a microphone of the user device, wherein the unknown audio track is associated with the media content emitting from a speaker. The media content is identified by acquiring identification data. The identification data is associated with the known media content at a known timepoint. If an audio tag is detected within the portion of the unknown audio track, identification data is extracted from the audio tag. If an audio tag is not detected within the portion of the unknown audio track, identification data is received from a server device.

[0011] Based on a user interaction with the user device, a request for content information from the server device is initiated. Upon receiving the request, content information associated with the identification data is transmitted from the server device to the user device. Transmitting content information comprises: receiving the request for content information from the user device, wherein the request comprises the identification data; using the identification data, retrieving stored content information associated with the identification data; and outputting the content information to the user device.

[0012] According to a further aspect of the present disclosure, there is provided a method for media content identification and content information retrieval at a user device. The method comprises monitoring a portion of an audio track using a microphone of the user device, wherein the audio track is associated with media content emitting from a speaker. It is then detected whether there is an audio tag embedded within the portion of the audio track, wherein the audio tag is embedded at frequencies detectable by the microphone and less detectable to a user of the user device. The audio tag comprises identification data associated with a known media content at a known timepoint.

[0013] If an audio tag is detected, the identification data is extracted. If an audio tag is not detected, identification data is received based on a best match for an audio fingerprint. The audio fingerprint comprises one or more audio features extracted from the portion of the audio track, and the best match is the most similar audio fingerprint from the plurality of audio fingerprints to the audio fingerprint. The media content associated with the audio track is then

identifiable based on the known media content of the identification data. The method further comprises retrieving content information associated with the identification data.

5 [0014] According to a further aspect of the present disclosure, there is provided a method for audio track identification and content information transmission at a server device. The method comprises generating a first audio fingerprint comprising one or more audio features extracted from a first portion of an audio track. The one or more audio features are either extracted from the first portion, which is obtained from a user device, or the one or more audio features are obtained from the user device. The method further comprises comparing the first audio fingerprint to a plurality of audio fingerprints. Each audio fingerprint of the plurality of audio fingerprints is associated with identification data, where the identification data is associated with a known audio track at a known timepoint.

10 [0015] A best match is then identified, wherein the best match is an audio fingerprint of the plurality of audio fingerprints that is most similar to the first audio fingerprint. The identification data associated with the best match is transmitted to the user device and, if a request is received from the user device, content information associated with the identification data is subsequently submitted to the user device.

15 [0016] According to a further aspect of the present disclosure, there is provided a method for method for subject extraction from media content at a workstation device. The method comprises obtaining a known media content, wherein the known media content comprises a known audio track. A timeline for the known media content is generated, wherein the timeline comprises a plurality of timepoints.

20 [0017] For the first time point of the timeline, one or more audio features is extracted from the known audio track at the first timepoint, and a first audio fingerprint comprising the one or more audio features is generated. Identification data associated with the known media content at the first timepoint is then linked to the first audio fingerprint. A first audio tag comprising the identification data is embedded into the known audio track at the first timepoint, generating an embedded audio track.

25 [0018] One or more subjects are extracted from the known media content, and information associated with the one or more subjects is collated to generate content information associated with the identification data. The method further comprises storing the first audio fingerprint, the embedded audio track, and the content information. The identification data is retrievable based on the first audio fingerprint or the embedded audio track, and the content information is retrievable based on the identification data.

30 [0019] According to a further aspect of the present disclosure, there is provided a method for preparing content interaction at a workstation platform. The method comprises importing a media content to the workstation platform, wherein the media content is associated with a plurality of subjects and a media content identifier. A timeline associated with the media content is generated, wherein the timeline comprises a plurality of timepoints. The workstation platform obtains one or more subject files, wherein each subject file is associated with a subject (of the plurality of subjects associated with the media content). Each subject file comprises a visual depiction of the subject text and/or a subject text, and is assigned to at least one timepoint on the timeline. An enriched media file comprising the timeline, the one or more subject files

35

40

associated with the timeline, and the media content identifier is then output from the workstation platform.

5 [0020] According to a further aspect of the present disclosure, there is provided a method for content interaction at a user platform. The method comprises monitoring an audio signal, wherein the audio signal comprises one or more audio tags embedded within the audio signal. A selected audio tag is obtained from the audio signal, which is used to identify a media content. Identification data is then extracted from the selected audio tag, wherein the identification data is associated with a media content identifier and a first timepoint.

10 [0021] If a lock-on indicator is not received at the user platform, the method further comprises obtaining a request indicator for a first subject file. The first subject file is associated with the media content identifier at the first timepoint (the identification data). The identification data is transmitted to a server device and the first subject file is subsequently received from the server device. The first subject file comprises a first visual depiction and a first subject text. Display of the first visual depiction or the first subject text is then initiated, and monitoring of the audio
15 signal continues.

[0022] If a lock-on indicator is received at the user platform, the method further comprises reducing monitoring of the audio signal. The identification data is transmitted to the server device and a plurality of subject files associated with the media content identifier of the identification data is requested. The plurality of subject files comprises the first subject file. The
20 plurality of subject files is received from the server, and display of the first visual depiction or the first subject text is initiated. When a second timepoint is reached, display of a second visual depiction or a second subject text from a second subject file associated with the second timepoint is initiated.

25 [0023] According to an additional aspect of the present disclosure, there is provided a computer-readable storage medium, which is transitory or non-transitory, comprising one or more program instructions which, when executed by one or more processors, cause the one or more processors to perform any of the methods disclosed herein.

[0024] Beneficially, the disclosure presents aspects of an innovative audio analysis system configured to revolutionize content discovery and interaction for media broadcasts. Unlike
30 conventional methods, the disclosed system allows users to identify and engage with media content without prior knowledge of its title or specifics. This groundbreaking technology not only identifies the content being played, but also synchronizes with it in real-time, providing a rich, interactive experience.

[0025] Aspects of the disclosed system and methods apply a dual-method approach to content
35 identification. The use of inaudible sound codes, audio tags, for direct identification, combined with the sophisticated audio property matching using audio fingerprints as a fallback, sets a new standard in the realm of audio analysis technologies. The integration of real-time error correction and environmental noise handling inherent to utilising audio tags or audio fingerprints demonstrates a significant advancement over traditional audio recognition
40 systems. Beneficially, this results in enhanced accuracy. By leveraging two complementary technologies, the systems and methods herein achieve higher accuracy in content identification, even in challenging environments. Further, this results in real-time interactivity.

Users can receive instantaneous information about the content, from basic details to in-depth data like specific products or scene descriptions. Further beneficially, this results in improved user experience. The seamless operation between the two audio-based approaches ensures a consistent and user-friendly experience, without manual intervention. The enclosed
5 technology has broad applications, from improving media identification and knowledge retrieval to offering real-time interaction experiences for subjects seen in these media. The in-built adaptability makes it suitable for various environments, whether in a quiet home setting or a noisy public space.

[0026] According to an aspect of the present disclosure, there is provided a method for secure
10 communication and output synchronisation with a transmission application at a user application. The method comprises scanning for content by broadcasting a first request using a bidirectional communication protocol. A microphone controllable from the user application detects a first sound code, where the first sound code is output from a speaker controllable from the transmission application. The first sound code is configured to be detectable by the
15 microphone and less detectable to a user of the user application. The method further comprises converting the first sound code into an identification code and

[0027] transmitting, using the bidirectional communication protocol, the identification code for authentication at the transmission application. A bidirectional connection for live data exchange is established with the transmission application, and a first output of the user application is
20 synchronised with a second output of the transmission application using the bidirectional connection. Beneficially, this method provides a secure and efficient way to pair and synchronise devices, enabling real-time data exchange and synchronisation of outputs, which enhances user experience and system security.

[0028] According to an additional aspect of the present disclosure, the bidirectional
25 communication protocol comprises WebSockets, and the bidirectional connection comprises a full-duplex connection. Beneficially, WebSockets and full-duplex connection enable continuous, real-time data exchange, reducing latency and improving synchronisation accuracy between devices.

[0029] According to an additional aspect of the present disclosure, the method further
30 comprises sustaining the bidirectional connection in response to detecting a second sound code within a predetermined interval after detecting the first sound code. Alternatively, the method further comprises terminating or otherwise closing the bidirectional connection in response to either (a) a user interaction with the user application, or (b) a location associated with the user application exceeding a predetermined threshold. Beneficially, this disclosure
35 ensures that the connection remains active only when necessary, enhancing security and resource management by preventing unauthorized access and reducing unnecessary data transmission.

[0030] According to an additional aspect of the present disclosure, the first sound code is an
40 ultrasound or near-ultrasound code. Beneficially, ultrasound codes are less detectable to users, providing a discreet and non-intrusive way to transmit data, further enhancing both user experience, accuracy of detection and resulting decoding, and system security.

[0031] According to an additional aspect of the present disclosure, the identification code comprises alphanumeric data, and converting the first sound code into the identification code comprises decoding the first sound code. Beneficially, this enhances accurate and secure identification of devices, reducing the risk of errors and unauthorized access.

5 [0032] According to an additional aspect of the present disclosure, detecting the first sound code is in response to the transmission application: receiving an identifier associated with the first request from a server; converting the identifier into the first sound code; and initiating output of the first sound code from the speaker. Beneficially, this results in secure and accurate transmission of pairing information, enhancing the reliability and security of the pairing process.

10 [0033] According to an additional aspect of the present disclosure, the method further comprises identifying a current timestamp associated with the first output and a message timestamp associated with the second output sent from the transmission application. A difference between the current timestamp and the message timestamp is reduced, thereby improving synchronicity between the first output and the second output. Beneficially, this
15 results in precise synchronisation of outputs, enhancing the user experience by providing seamless and synchronised content across devices and improving security by maintaining temporal alignment across devices.

[0034] According to an aspect of the present disclosure, there is provided a method for secure communication and output synchronisation with a user application at a transmission
20 application. The method comprises receiving, using a bidirectional communication protocol, a first request associated with the user application. A pairing request is transmitted to a server, wherein the pairing request comprises data associated with the first request. If the server identifies the pairing request, an identifier is received from the server. The transmission application converts the identifier into a first sound code for emission from a speaker, wherein
25 the speaker is controllable from the transmission application. The method further comprises receiving, using the bidirectional communication protocol, an identification code from the user application and authenticating the identification code based on matching the identifier and the identification code. If the identification code is authenticated, a handshake is performed, establishing a bidirectional connection for live data exchange with the user application. A first
30 output of the user application is synchronised with a second output of the transmission application using the bidirectional connection. Beneficially, this disclosure provides a secure and efficient way to pair and synchronise devices, enabling real-time data exchange and synchronisation of outputs, which enhances user experience and security.

[0035] According to an additional aspect of the present disclosure, the bidirectional
35 communication protocol comprises WebSockets, and the bidirectional connection comprises a full-duplex connection. Beneficially, WebSocket and full-duplex communication enables continuous, real-time data exchange, reducing latency and improving synchronisation accuracy between devices.

[0036] According to an additional aspect of the present disclosure synchronisation comprises
40 receiving, using the bidirectional connection, a message timestamp associated with the first output. A current timestamp associated with the second output is determined and synchronisation is based on reducing a difference between the message timestamp and the current timestamp, thereby improving synchronicity between the first output and the second

output. Beneficially, this enables precise synchronisation of outputs, enhancing security and the user experience by providing seamless and synchronised content across devices.

5 [0037] According to an aspect of the present disclosure, there is provided a transitory or non-transitory computer-readable medium storing instructions for performing the method of any of the aspects described above. Beneficially, this medium provides a convenient and efficient way to implement the described methods, ensuring consistency and reliability in the execution of the pairing and synchronisation processes.

10 [0038] According to an aspect of the present disclosure, there is provided a system for secure communication and output synchronisation between a user device and a transmission device. The system comprises a user device and a transmission device configured to perform the above methods. Beneficially, this system provides a comprehensive solution for secure communication and synchronisation between devices, enhancing user experience and security.

15 [0039] According to an aspect of the present disclosure, there is provided a Software Development Kit (SDK) for secure communication and output synchronisation. The SDK comprises one or more communication modules, one or more authentication modules, and one or more interaction modules. The one or more communication modules are configured to: send data from a transmission device to a user device using a bidirectional protocol; receive data from a user device to the transmission device using the bidirectional protocol; establish a
20 bidirectional connection between the transmission device and the user device; transmit data to a server; receive data from the server; and transmit instructions to a speaker. The one or more authentication modules are configured to: generate a pairing request associated with data received from the user device; generating a first sound code by encoding an identifier received from the server; and compare the identification code to the identifier received from
25 the server. The one or more interaction modules are configured to: identify a current timestamp associated with an output of the transmission device; calculate a difference between a message timestamp received from the user device and the current timestamp; and

[0040] initiate adjustment of the output of the transmission device to minimise the difference. Beneficially, the SDK provides a robust and flexible framework for implementing secure
30 communication and synchronisation, enhancing the development and deployment of such systems.

[0041] According to an additional aspect of the present disclosure, the one or more communication modules are configured to establish a secure WebSocket connection (WSS) using a TLS/SSL protocol. Beneficially, this enables secure data transmission, protecting
35 against unauthorized access and data breaches, thereby enhancing the security of the communication process.

[0042] According to an additional aspect of the present disclosure, generating the first sound code further comprises encoding the current timestamp with added redundancy to facilitate error detection and correction during transmission, emission, playback, or decoding
40 processes. Beneficially, this improves the reliability and accuracy of data transmission, reducing the risk of errors and enhancing the overall performance of the system.

[0043] In summary, the above aspects leverage inaudible sound codes for device pairing and synchronisation, which are less detectable to users but easily captured by device microphones. This approach enhances security and user convenience by automating the pairing process and reducing the need for manual input.

5 [0044] The above aspects provide significant technical benefits over existing systems, particularly in the context of pairing and synchronisation of multimedia content. The disclosed methods and systems automate the pairing process through sound code detection, eliminating the need for manual input such as entering PINs or scanning QR codes. This automation simplifies the user experience, making it more intuitive and less error-prone, allowing users to
10 effortlessly connect their devices without navigating through complex setup procedures. Additionally, the system ensures real-time synchronisation of outputs between devices by establishing a bidirectional connection for live data exchange. The use of timestamps to synchronise outputs addresses issues related to latency, network jitter, and varying processing times, resulting in a more cohesive and immersive user experience, especially in multimedia
15 applications where audio and video synchronisation is beneficial.

[0045] The ability to sustain or terminate the bidirectional connection based on sound code detection, user interaction, or location thresholds provides dynamic management of connections. This adaptability ensures continuous secure communication and synchronisation, even in changing conditions, enhancing the reliability and robustness of the system.
20 Furthermore, the inclusion of a Software Development Kit (SDK) and computer-readable media storing instructions for the method provides developers with the tools needed to implement secure communication and synchronisation in their applications. This facilitates widespread adoption and integration of the technology, promoting a more secure and synchronised ecosystem of connected devices. The use of bidirectional communication protocols, such as
25 WebSockets, further ensures secure data transmission, enhancing the overall security and efficiency of the system.

[0046] According to another aspect of the present disclosure, there is provided a method for maintained media content synchronisation at a user device, including receiving a root hash of a hierarchical hash structure associated with media content. Higher hierarchical levels of the
30 hash structure comprise combined hash values from lower levels, culminating in the root hash. The system retrieves data from a database associated with the root hash, including content information for media content interaction and a contextual tag for differentiating between known media contents, e.g. between standalone songs and the same songs within a movie soundtrack. The media content is monitored at a first timestamp for a predetermined duration,
35 obtaining a first media segment. Synchronisation is verified by computing a hash of the first media segment and comparing it with the hierarchical hash structure. If synchronisation is verified, the hash is used to retrieve and display content information associated with the first timestamp. If not, the system adjusts the timestamp for resynchronisation or uses a machine learning model to generate a predicted hash for retrieving content information. The media
40 content is then monitored at a second timestamp within the predetermined duration from the first timestamp, obtaining a second media segment overlapping with the first. A hash of the second media segment is computed to maintain synchronisation.

[0047] Beneficially, by leveraging a hierarchical hash structure, the method ensures robust and precise synchronisation of media content. This method allows for real-time verification of synchronisation by computing and comparing hashes of media segments, which enhances the accuracy and reliability of content interaction. The ability to adjust timestamps and use machine learning models for generating predicted hashes ensures that synchronisation is maintained even in dynamic or disrupted environments. Additionally, the method's capability to switch between hash-based synchronisation and audio tag detection provides flexibility and resilience, accommodating various media content types and playback conditions. This dual approach not only improves the user experience by providing seamless and uninterrupted content interaction but also ensures that content information is accurately retrieved and displayed, ensuring secure and accurate synchronisation of data between devices.

[0048] It is understood that the described apparatus, process, system, and method are not limited to pairing and synchronization of multimedia content and may be applied to other contexts and usage scenarios. For example, the described technology can be used in secure access control systems, where ultrasound codes are used to authenticate users and grant access to restricted areas. Additionally, it can be applied in industrial automation for synchronising operations between different machines or devices, ensuring precise coordination and timing. The technology can also be adapted for use in smart home environments, where various smart devices need to communicate and synchronise their actions seamlessly.

BRIEF DESCRIPTION OF DRAWINGS

[0049] Embodiments of the invention will now be described, by way of example only, and with reference to the accompanying drawings, in which:

[0050] Figure 1 illustrates a system architecture diagram for a media interaction system;

[0051] Figure 2 illustrates a system architecture diagram for processing media content at a workstation device;

[0052] Figures 3A and 3B illustrate a system architecture diagram for identifying and synchronizing with media content at a user device;

[0053] Figure 4 illustrates a system architecture diagram for identifying media content and retrieving content information;

[0054] Figure 5 illustrates a system architecture diagram for pairing initiation;

[0055] Figure 6 illustrates a system architecture diagram for synchronising output;

[0056] Figure 7 illustrates a system architecture diagram for pairing maintenance;

[0057] Figure 8 illustrates a system architecture diagram for identification retrieval;

[0058] Figure 9 illustrates a user interface sequence for initiating content interaction;

[0059] Figure 10 illustrates a user interface sequence for terminating content interaction;

[0060] Figure 11 illustrates a system architecture diagram for an SDK;

[0061] Figure 12 illustrates a flowchart of a method for performing media interaction;

[0062] Figure 13 illustrates a flowchart of a method for preparing content interaction at a workstation platform;

[0063] Figures 14A and 14B illustrate a flowchart of a method for subject extraction from media content at a workstation device;

[0064] Figure 15 illustrates a flowchart of a method for media content identification and content information retrieval at a user device;

[0065] Figures 16A and 16B illustrate a flowchart of a method for content interaction at a user platform;

5 [0066] Figure 17 illustrates a flowchart of a method for audio track identification and content information transmission at a server device;

[0067] Figures 18A, 18B and 18C illustrate a flowchart of a method for media content identification and interaction;

10 [0068] Figure 19 illustrates a flowchart of a method for establishing secure communication at a user device;

[0069] Figure 20 illustrates a flowchart of a method for establishing secure communication at a transmission application;

[0070] Figures 21A and 21B illustrate a system architecture diagram for a video interaction system;

15 [0071] Figure 22 illustrates a system architecture diagram of a dual synchronisation system;

[0072] Figure 23 illustrates a flowchart of method for processing video content; and

[0073] Figure 24 illustrates a flowchart of a method for synchronising based on video hashing; and

20 [0074] Figure 25 shows an example computing environment for performing any of the methods described herein.

[0075] All illustrations of the drawings are for the purpose of describing selected versions of the present invention and are not intended to limit the scope of the present invention.

25 [0076] As a preliminary matter, it will readily be understood by one having ordinary skill in the relevant art that the present disclosure has broad utility and application. As should be understood, any embodiment may incorporate only one or a plurality of the above-disclosed aspects of the disclosure and may further incorporate only one or a plurality of the above-disclosed features. Furthermore, any embodiment discussed and identified as being “preferred” is considered to be part of a best mode contemplated for carrying out the embodiments of the present disclosure.

30 [0077] Other embodiments also may be discussed for additional illustrative purposes in providing a full and enabling disclosure. Moreover, many embodiments, such as adaptations, variations, modifications, and equivalent arrangements, will be implicitly disclosed by the embodiments described herein and fall within the scope of the present disclosure. Accordingly, while embodiments are described herein in detail in relation to one or more embodiments, it is
35 to be understood that this disclosure is illustrative and exemplary of the present disclosure, and are made merely for the purposes of providing a full and enabling disclosure.

40 [0078] The detailed disclosure herein of one or more embodiments is not intended, nor is to be construed, to limit the scope of patent protection afforded in any claim of a patent issuing here from, which scope is to be defined by the claims and the equivalents thereof. It is not intended that the scope of patent protection be defined by reading into any claim a limitation found herein that does not explicitly appear in the claim itself. Additionally, it is beneficial to note that each term used herein refers to that which an ordinary artisan would understand such term to mean based on the contextual use of such term herein. To the extent that the meaning

of a term used herein—as understood by the ordinary artisan based on the contextual use of such term—differs in any way from any particular dictionary definition of such term, it is intended that the meaning of the term as understood by the ordinary artisan should prevail. Furthermore, it is beneficial to note that, as used herein, “a” and “an” each generally denotes
5 “at least one,” but does not exclude a plurality unless the contextual use dictates otherwise. When used herein to join a list of items, “or” denotes “at least one of the items,” but does not exclude a plurality of items of the list. Finally, when used herein to join a list of items, “and” denotes “all of the items of the list.”

[0079] The following detailed description refers to the accompanying drawings. Wherever
10 possible, the same reference numbers are used in the drawings and the following description to refer to the same or similar elements. While many embodiments of the disclosure may be described, modifications, adaptations, and other implementations are possible. For example, substitutions, additions, or modifications may be made to the elements illustrated in the drawings, and the methods described herein may be modified by substituting, reordering, or
15 adding stages to the disclosed methods. Accordingly, the following detailed description does not limit the disclosure. Instead, the proper scope of the disclosure is defined by the appended claims.

[0080] The present disclosure contains headers. It should be understood that these headers are used as references and are not to be construed as limiting upon the subjected matter
20 disclosed under the header. Other technical advantages may become readily apparent to one of ordinary skill in the art after review of the following figures and description. It should be understood at the outset that, although exemplary embodiments are illustrated in the figures and described below, the principles of the present disclosure may be implemented using any number of techniques, whether currently known or not. The present disclosure should in no
25 way be limited to the exemplary implementations and techniques illustrated in the drawings and described below.

[0081] Unless otherwise indicated, the drawings are intended to be read together with the specification, and are to be considered a portion of the entire written description of this invention. As used in the following description, the terms “horizontal”, “vertical”, “left”, “right”,
30 “up”, “down” and the like, as well as adjectival and adverbial derivatives thereof (e.g., “horizontally”, “rightwardly”, “upwardly”, “radially”, etc.), simply refer to the orientation of the illustrated structure as the particular drawing figure faces the reader. Similarly, the terms “inwardly”, “outwardly” and “radially” generally refer to the orientation of a surface relative to its axis of elongation, or axis of rotation, as appropriate.

[0082] The present disclosure includes many aspects and features. Moreover, while many aspects and features relate to, and are described in the context of a media content interaction or synchronisation system, embodiments of the present disclosure are not limited to use only
35 in this context. In the context of the present invention any systems, methods or processes disclosed herein comprise an at least one processing unit whereby said at least one processing unit performs the process of the present invention.
40

DETAILED DESCRIPTION

[0083] Embodiments of the present disclosure will now be described with reference to the attached figures. It is to be noted that the following description is merely used for enabling the skilled person to understand the present disclosure, without any intention to limit the applicability of the present disclosure to other embodiments which could be readily understood and/or envisaged by the reader. In particular, whilst the present disclosure is primarily directed to an audio recognition and content identification system for enhanced media interaction, the skilled person will appreciate that the methods and systems described herein are applicable to audio signal-based identification and secure synchronisation, and relevant data retrieval, more generally.

[0084] One advantage of the systems, methods, and technology described herein is to bridge the gap between broadcast media and user interactivity. The enclosed embodiments, e.g. embodiments associated with Figures 1 to 4, improve identification of media content and data acquisition, resulting in increases to efficiency, security, and reliability of data retrieval over known methods. For example, viewers can identify the exact frame of a movie playing on TV and access related information, including actors, locations, and products featured in the scene, such as clothing or decor items. Multiple interactions between a viewer/user and a user device are no longer required, e.g. as were previously needed for searching for the correct movie, searching for the correct scene, and then further searching for the related information. Such a reduction not only improves man-machine interaction, with more accurate, time efficient and secure data retrieval, but also presents an improvement to energy efficiency and battery life of the user device.

[0085] Another advantage of the disclosed embodiments is a dual-method process for content identification and synchronization. The first method involves embedding unique, preferably inaudible, sound codes within the audio track of movies and broadcasts as audio tags. These audio tags, detectable by a microphone of a user device, comprise information associated with the content and its current playback position. Should factors impede the first method, such as environmental factors including background noise, there is a reliable fallback mechanism based on audio property analysis. This second method ensures continuity in content identification and synchronization. A combination of technologies, a dual-method approach, provides a seamless and enriched content interaction experience, offering users immediate access to detailed information about the content they are watching, e.g. down to specific scenes and items within those scenes.

[0086] There is a growing need for a more intuitive, efficient, and interactive approach to content discovery and engagement. Viewers today seek instant identification and deeper connections with the content, desiring information that extends beyond basic details to include real-time data, contextual insights, and interactive elements like product details and scene-specific facts. Additionally, as user devices become more efficient, man-machine interactions need to be reduced and the efficiency of overall data retrieval systems increased. Moreover, reduced man-machine interactions and improved accuracy and reliability result in a safer more secure data retrieval system, as not only is the retrieved information relevant and e.g. cybersecure, but risks of malware or malicious content retrieval is effectively eliminated.

5 [0087] Additionally, whilst the present disclosure is primarily directed to pairing and synchronisation of multimedia content, the skilled person will appreciate that the apparatus, processes, methods, and systems described herein are applicable to various other fields. For example, the technology can be utilized in secure access control systems, where ultrasound codes are employed to authenticate users and grant access to restricted areas. Additionally, it can be adapted for industrial automation to synchronise operations between different machines or devices, ensuring precise coordination and timing. Furthermore, the technology can be applied in smart home environments, enabling seamless communication and synchronisation among various smart devices.

10 [0088] Benefits of the enclosed systems and methods include enhanced security, seamless user experience, real-time synchronisation, dynamic connection management, and developer-focused efficient implementation. Overall, the proposed method offers substantial technical benefits over existing systems, addressing beneficial issues in security, user experience, synchronisation, and ease of implementation.

15 [0089] By automating the pairing process through sound code detection, the method eliminates the need for manual input, such as entering PINs or scanning QR codes. This automation simplifies the user experience, making it more intuitive and less error prone. Users can effortlessly connect their devices without navigating through complex setup procedures.

20 [0090] The method and systems ensure real-time synchronisation of outputs between devices by establishing a bidirectional connection for live data exchange. The use of timestamps to synchronise outputs addresses issues related to latency, network jitter, and varying processing times. This results in a more cohesive and immersive user experience, particularly in multimedia applications, e.g. apps, where audio and video synchronisation is beneficial.

25 [0091] Additionally, the ability to sustain or terminate the bidirectional connection based on sound code detection, user interaction, or location thresholds provides dynamic management of connections. This adaptability ensures continuous secure communication and synchronisation, even in changing conditions, enhancing the reliability and robustness of the system.

30 [0092] Furthermore, the inclusion of a Software Development Kit (SDK) and computer-readable media storing instructions for the method provides developers with the tools needed to implement secure communication and synchronisation in their applications. This facilitates widespread adoption and integration of the technology, promoting a more secure and synchronised ecosystem of connected devices.

35 [0093] Methods provided herein significantly improve security by utilizing sound codes, particularly ultrasound or near-ultrasound codes, for device pairing and authentication. These sound codes are encoded and decoded at authorised devices and/or using authorised applications or SDKs, reducing the risk of unauthorized access and man-in-the-middle attacks that are common in traditional methods like Bluetooth pairing or QR code scanning. The use of bidirectional communication protocols, such as WebSockets, further ensures secure data
40 transmission.

[0094] Figure 1 illustrates a system architecture diagram for a media interaction system. Specifically, Figure 1 shows a block diagram of an embodiment of a media interaction system 100 according to example embodiments of the present disclosure.

5 [0095] Media interaction system 100 comprises: known media content 102, workstation 104, user device 106, content information 108, server 110, media content source 112, database 114, embedded audio track 116, audio track portion 118, speaker 120, identification data 122, and request 124. Media content source 112 comprises media content storage 112-A and media content player 112-B. Audio track portion 118 comprises an audio signal 118-A and is associated with unidentified media content 118-B. Optionally, server 110 comprises database
10 114.

[0096] In one example, the media interaction system 100 is configured to: process known media content 102 at workstation 104; facilitate interaction with the known media content 102 at user device 106; and transmit content information 108 from server 110 to user device 106. Workstation 104 obtains known media content 102 from media content source 112 or database
15 114, and stores embedded audio track 116 and content information 108 at database 114 or server 110. The user device 106 monitors audio track portion 118 emitting from speaker 120 and acquires identification data 122 either using audio track portion 118 or from server 110. Request 124 is initiated to server 110 based on a user interaction with user device 106. Server 110 receives request 124 and uses identification data 122 to retrieve stored content
20 information 108. Content information 108 is then output to user device 106.

[0097] Workstation 104 is a computing system, device or software application configured to process known media content 102. Known media content 102 comprises any media content associated with audio, such as television shows, movies or films, audiobooks, podcasts, music, video games, social media content, web series, streamed media or broadcast media.
25 Therefore, known media content 102 comprises a known audio track. Known media content 102 is obtained by workstation 104 from media content source 112 or database 114.

[0098] Media content source 112 is media content storage 112-A or media content player 112-B, where media content storage 112-A is optionally database 114, server 110, user device 106, or another source of known media content 102 e.g. a device or platform external from media
30 interaction system 110 such as a device associated with the owner, creator, distributor, or license holder of the known media content 102. Media content player 112-B is a source of known media content 102 and optionally unidentified media content 118-B. For example, media content player 112-B is a media broadcaster, a streaming provider, or a device capable of playing, streaming or storing media content, such as a device comprising speaker 120.
35 Optionally, media content source 112 stores known media content 102 in database 114, where known media content 102 is subsequently retrievable at workstation 104.

[0099] Workstation 104 is configured to generate a timeline from the known media content 102, generate embedded audio track 116, and generate content information 108, e.g. as shown in relation to Figure 2 below. Embedded audio track 116 comprises one or more audio tags,
40 which are preferably embedded at frequencies detectable at user device 106 but beneficially inaudible to a user of user device 106. The audio tags comprise identification data 122, and hence embedded audio track 116 comprises identification data 122. Content information 108 is linked to identification data 122, as content information 108 comprises collated information

associated with one or more subjects extracted from known media content 102. Subjects are objects, locations, people, services, or other datapoints relevant to the known media content 102. Optionally, workstation 104 comprises a machine learning model to identify and/or classify objects in known media content 102.

5 [0100] User device 106 is configured to monitor audio track portion 118 emitting from speaker 120, such as by detecting, measuring, and/or collecting audio signal 118-A using a microphone, or by otherwise observing unidentified media content 118-B. For example, user device 106 is a mobile phone, a software application, a personal tablet or computer, or any other platform or device communicatively or electronically coupled to a microphone. The
10 unidentified media content 118-B associated with the audio track portion 118 is identifiable based on identification data 122. The identification data 122 comprises a media identifier (or other indicator of a specific media content e.g. the known media content 102) and an indicator of a specific time or time interval within the specific media content, e.g. a timepoint of the timeline generated by workstation 104. If an audio tag is detected by user device 106,
15 identification data 122 is extracted from the audio tag. If an audio tag is not detected by user device 106, e.g. if audio track portion 118 is not embedded with an audio tag or if there are factors impacting detection of audio tags such as environmental noise, identification data 122 is obtained from server 110.

[0101] For example, server 110 obtains one or more audio features extracted from audio track
20 portion 118, either directly from user device 106 or by obtaining audio track portion 118 from user device 106 and extracting the one or more audio features. An unknown audio fingerprint is then generated comprising the one or more audio features, and the unknown audio fingerprint is compared to a plurality of audio fingerprints e.g. stored at database 114. The most similar audio fingerprint in the plurality of audio fingerprints to the unknown audio fingerprint is
25 a best match, and information data associated with the best match is then transmitted to user device 106. Optionally, the plurality of audio fingerprints is generated by workstation 104, where workstation 104 is further configured to generate an audio fingerprint for each timepoint of the timeline and link identification data to each audio fingerprint.

[0102] Server 110 is configured to receive request 124, where outputting of request 124 from
30 user device 106 is initiated based on a user interaction with user device 106. For example, the user indicates that they would like to receive content information 108 associated with audio track portion 118 at user device 106. Request 124 comprises identification data 122, such that server 110 retrieves content information 108 associated with identification data 122 from the database 114. Optionally, audio signal 118-A is associated with a plurality of known media
35 content, such that the user interaction with user device 106 comprises selecting identification data 122 from a plurality of identification data.

[0103] Beneficially, user device 106 is independent of audio track portion 118, meaning that user device 106 does not need to be in communication with media content source 112 to identify audio track portion 118 or receive content information associated with audio track
40 portion 118. Additionally, speaker 120 does not need to emit embedded audio track 116 for the user device 106 to identify audio track portion 118, as user device 106 can identify and obtain content information 108 relevant to played media content that does not comprise audio tags generated by workstation 104. Provided an audio fingerprint has been generated that is

associated with audio track portion 118, the audio track portion 118 is identifiable by media interaction system 100.

[0104] Another benefit of media interaction system 100 is secure and efficient content information 108 retrieval. A single user interaction at user device 106 results in receiving content information 108 that is relevant to audio track portion 118 and from a safe and trusted source, i.e. server 110 and/or database 114. Therefore, there is a reduced risk of malware or malicious data being received at the user device 106 and an increase in efficiency and battery life associated with user device 106. Unlike alternative technologies, the user device does not (a) have to be linked or otherwise is communication with media content player 112-B to receive relevant content information or (b) have to perform extensive search operations with either multiple interactions between the user and the user device or application of excessive computational resources in order to automate such search operations, e.g. use of complex artificial intelligence algorithms.

[0001] Figure 2 illustrates a system architecture diagram for processing media content at a workstation device. Specifically, Figure 2 shows a block diagram of an embodiment of a media processing system 200 according to example embodiments of the present disclosure.

[0105] Media processing system 200 comprises: workstation 202, known media content 204, audio track 206, server 208, timeline 210, first timepoint 212, audio features 214, audio fingerprint 216, content identifier 218, embedded audio track 220, audio tag 222, subjects 224, content file 226, and, optionally, enriched media file 228.

[0106] In one example, media processing system 200 comprises workstation 202 configured to obtain known media content 204, where the known media content comprises an audio track 206. The known media content 204 is obtained from server 208, a database, or another media content source not depicted. A timeline 210 is generated for the known media content 204, the timeline comprising a plurality of timepoints including the first timepoint 212. Audio features 214 are extracted from the audio track 206 to generate an audio fingerprint 216 linked to identification data comprising media content identifier 218 and the first timepoint 212. Audio fingerprint 216 is generated at workstation 202, server 208, or another device not depicted. Media processing system is also configured to generate an embedded audio track 220 by embedding the audio tag 222 within the audio track 206, where the audio tag 222 comprises the identification data. Embedded audio track 220 is generated at either workstation 202 or server 208. The audio fingerprint 216, the embedded audio track 220 are stored at server 208, workstation 202, or a database, e.g. database 114 of Figure 1.

[0107] Additionally, subjects 224 are extracted from the known media content 204 and content file 226 is obtained, where content file 226 comprises a visual depiction of subject 224-A and a subject text associated with subject 224-A. Enriched media file 228 is output to the server 208, a database, or workstation 202, where the enriched media file 228 is stored, e.g. for access from server 208 or for content file 226 retrieval for a user device, or modified/edited, e.g. at workstation 202. The enriched media file 228 comprises the timeline 210, content file 226, and media content identifier 218.

[0108] Figures 3A and 3B illustrate a system architecture diagram for identifying and synchronizing with media content at a user device. Specifically, Figures 3A and 3B show a

block diagram of an embodiment of a content synchronisation system 300 according to example embodiments of the present disclosure.

[0109] Content synchronisation system 300 comprises: audio signal 302, speaker 304, first audio tag 306, second audio tag 308, user device 310, user interface 310-A, identification data 312, server 314, first subject file 316, database 318, first visual depiction or subject text 320, second subject file 322, second visual depiction or subject text 324, first audio features 326, and second audio features 328.

[0110] In one example, content synchronisations system 300 is configured to monitor audio signal 302 emitting from speaker 304, where first audio tag 306 and second audio tag 308 are embedded in audio signal 302, using a microphone or other audio sensor of user device 310. User device 310 extracts identification data 312 from first audio tag 306. When user interface 310-A receives a request indicator, which is indicative of a request for subject information associated with audio signal 302, identification data 312 from first audio tag 306 is transmitted to server 314. First subject file 316 is retrieved by server 314, e.g. from database 318, and received at user device 310. First subject file 316 comprises first visual depiction or subject text 320, which can be displayed at user device 310. In order to synchronise displayed subject information, monitoring of audio signal 302 continues and, for example, user device 310 extracts identification data 312 from second audio tag 308. When user interface 310-A receives a request indicator, identification data 312 from second audio tag 308 is transmitted to server 314. Second subject file 322 is retrieved by server 314 and received at user device 310. Second subject file 322 comprises second visual depiction or subject text 324, which can be displayed at user device 310.

[0111] In another example, content synchronisations system 300 is configured to monitor audio signal 302 emitting from speaker 304, where first audio tag 306 and second audio tag 308 are not embedded in audio signal 302 or are otherwise undetectable using a microphone or other audio sensor of user device 310. User device 310 extracts first audio features 326 from audio signal 302 at a first timepoint, or collects a first portion of the audio signal 302 at the first timepoint. To identify the audio signal 302, user device 310 transmits first audio features 326 or the first portion to the server 314. Server 314 retrieves identification data 312 using the first audio features 326, which is transmitted to the user device 310. When user interface 310-A receives a request indicator, which is indicative of a request for subject information associated with audio signal 302, identification data 312 is transmitted to server 314. First subject file 316 is retrieved by server 314, e.g. from database 318, and received at user device 310. First subject file 316 comprises first visual depiction or subject text 320, which can be displayed at user device 310. Optionally, user device 310 extracts second audio features 328 from audio signal 302 at a second timepoint, or collects a second portion of the audio signal 302 at the second timepoint. To synchronise display of content information with the audio signal 302, user device 310 transmits second audio features 328 or the first portion to the server 314. Server 314 retrieves identification data 312 using the second audio features 328, which is transmitted to the user device 310. When user interface 310-A receives a request indicator, identification data 312 is transmitted to server 314. Second subject file 322 is retrieved by server 314 and received at user device 310. Second subject file 322 comprises second visual depiction or subject text 324, which can be displayed at user device 310.

[0112] Optionally, a lock-on indicator is received, as described in relation to method 800 below.

[0113] Figure 4 illustrates a system architecture diagram for identifying media content and retrieving content information. Specifically, Figure 4 shows a block diagram of an embodiment of an audio recognition system 400 according to example embodiments of the present disclosure.

[0114] Audio recognition system 400 comprises: server 402, user device 404, first portion 406, audio features 408, first audio fingerprint 410, plurality of audio fingerprints 412, database 414, identification data 416, request 418, content information 420, and relevant information 422. Plurality of audio fingerprints 412 comprises second audio fingerprint 412-A, third audio fingerprint 412-B, and best match 412-C. Relevant information 422 comprises relevant information 422-A associated with second audio fingerprint 412-A, relevant information 422-B associated with second audio fingerprint 412-B, and relevant information 422-C associated with best match 412-C.

[0115] In one example, audio recognition system 400 comprises server 402 configured to obtain, from user device 404, a first portion 406 of an audio track or audio features 408 extracted from the first portion 406. If first portion 406 is not received, server 402 extracts audio features 408 from the first portion 406. First audio fingerprint 410 is generated from audio features 408, and compared to a plurality of audio fingerprints 412 stored at database 414. Best match 412-C is the most similar audio fingerprint of the plurality of audio fingerprints 412 to first audio fingerprint 410. Identification data 416 associated with best match 412-C is retrieved by server 402 and transmitted to user device 404. If a request 418 for content information 420 is received from user device 404, content information 420 is transmitted to the user device 404, where content information 420 is relevant information 422-C associated with best match 412-C and identification data 416.

[0116] Relevant information 422 comprises information associated with one or more subjects relevant to the first portion 406 or, specifically, one or more subjects relevant to a media content associated with the audio track at the time of the first portion 406, e.g. first subject file 316 of Figure 3A or content file 226 of Figure 2.

[0117] Optionally, server 402 or user device 404 is configured to adaptively optimize audio processing parameters based on the detected media format and transmission quality, enhancing the resilience of the technology to signal degradation. Therefore, while media formats and transmission mediums can cause degradation of the audio signal, audio recognition system 400 is configured to reduce any resulting impact from varied media formats and transmission mediums. Further optionally, audio recognition system 400 is configured to ensure beneficially real-time processing of audio data and synchronization between the server 402, the database 414 (optionally a server database and comprised within server 402), and the user device 404 without requiring inefficient levels of significant computational resources. This is achieved by optimisation of data processing workflows and implementation of efficient algorithms allowed for real-time operation without compromising accuracy or user experience.

[0118] For example, machine learning algorithms are incorporated to improve and optimise system performance based on user interactions and feedback. In one example, a machine learning algorithm is used to determine the best match 412-C. E.g., a machine learning

algorithm is used to generate a first audio fingerprint 410 or identify an audio tag in first portion 406. In another example, data caching strategies are employed to reduce server response times, enhancing the overall speed of content identification and information retrieval.

5 [0119] Further optionally, audio recognition system 400 comprises a transcript database (not depicted). The transcript database comprises a plurality of transcribed media content, allowing for word-based content identification and synchronisation. For example, the user device 404 is configured to transmit spoken or typed queries to the server 402, such as quoting a line from a movie or describing a scene, leading to identification of the specific media content using the transcript database.

10 [0120] In one example, the artificial intelligence algorithms are integrated at the user device 404 and are configured to interpret and analyse spoken and/or typed user queries by processing words, phrases, or descriptions provided by the user. The artificial intelligence algorithm then cross-references these queries, via the server 402, with the transcript database, which further stores metadata comprising any of titles, actor names, thematic elements, or
15 other data relevant for identifying a best match. These trained artificial intelligence algorithms are updated based on user interaction with the user device 404, such that the artificial intelligence algorithms become personalised to users or the user device 404. For example, media content identification or retrieval of relevant information associated with the media content is improved based on e.g. individual user preferences or viewing histories.

20 [0121] In another example, the artificial intelligence algorithm learns from user interactions, enhancing accuracy and efficiency of identification and information retrieval. E.g., the user device 404 receives whether the best match 412-C is or is not a successful match to the media content emitting from the speaker, which is used to improve the artificial intelligence algorithm for identifying a best match from database 414.

25 [0122] More specifically, an example artificial intelligence algorithm is a support vector machine (SVM), and another is an artificial neural network (ANN). The SVM is a supervised learning model used for classification and regression tasks. For identifying best match 412-C within database 414, the SVM is trained on the plurality of audio fingerprints, which store information related to audio characteristics such as frequency distribution, rhythm patterns, or
30 spectral characteristics. The training data consists of pairs of input audio fingerprints and corresponding labels indicating the most similar audio fingerprint in the database. Once trained, the SVM takes an audio fingerprint input and classifies it to identify the most similar audio fingerprint based on learned patterns during training. The ANN is a deep neural network trained on a dataset similar to that of the SVM. The ANN would typically consist of multiple
35 layers of neurons, with techniques like backpropagation used for training. Once trained, the ANN is input an audio fingerprint, which is passed through its layers, and outputs a prediction indicating the most similar audio fingerprint from the database 414 based on the learned representations. In both cases, the choice between using e.g. an SVM or an ANN depends on factors such as the complexity of the data, the size of the dataset, and computational resources
40 available. ANNs, particularly deep learning models, might offer better performance for tasks involving large datasets and complex patterns, but they also require more computational resources and data to train effectively. SVMs, on the other hand, are simpler models that can work well with smaller datasets and may require less computational overheads. Preferably, the

choice of artificial intelligence algorithm is tailored to the computational resources and training data available to the audio recognition system 400.

5 [0123] Preferably, server 402 and database 414, which is optionally integrated with server 402, is cloud-based. Beneficially, a mixture of cloud-based infrastructure and distributed computational resources as shown in Figures 1-4 results in increased reliability and scalability. In an example where the server is cloud-based, choice of artificial intelligence model includes use of large pre-trained generative algorithms, which are then tailored using e.g. the training data described above to perform any of the method steps described in relation to methods 500-1100 below.

10 [0124] The Figures 5 to 11 described in detail below are associated with secure communication and output synchronisation between a user device and a transmission device. According to one embodiment, a user device scans for content, such as content being displayed on a transmission device. The transmission device communicates with a server to identify the request and, if identified by the server, generate a sound code. The sound code
15 comprises an encoded text code used for authentication of the communication between the user device and the transmission device. The user device detects the sound code, generating a decoded text code, which is then transmitted to the transmission device. If the pre-encoded text code and decoded text code match, the communication is authenticated, and a handshake is performed. The handshake establishes a bidirectional connection for live data exchange
20 between the transmission device and the user device, resulting in efficient and secure communication between devices for content interaction. This bidirectional connection is then used for output synchronisation, resulting in improved user experience, enhanced accuracy, reduced latency, increased reliability, improved system security, resource efficiency, and versatility.

25 [0125] Synchronising output between a user application and a transmission application ensures seamless content display across multiple devices, eliminating lag and discrepancies for a more cohesive viewing experience. Using timestamps for synchronisation achieves high precision, which is beneficial for live streaming, gaming, and interactive multimedia. Real-time data exchange via WebSockets minimizes latency, ensuring immediate reflection of actions
30 across devices. Periodic heartbeat messages and error correction mechanisms enhance stability and reliability, even in noisy environments. The use of ultrasound or near-ultrasound codes for pairing adds a security layer, while secure WebSocket connections using TLS/SSL protocols protect data transmission. Ultrasound or near-ultrasound codes for device pairing and authentication significantly improves security by reducing the risk of unauthorized access
35 and man-in-the-middle attacks, which are common in traditional methods like Bluetooth pairing or QR code scanning. Additionally, the ultrasound or near ultrasound frequencies, or alternative frequencies for implementation, represent frequencies with less interference, resulting in more accurate detection and decoding. Efficient resource management is achieved by sustaining the bidirectional connection only when necessary, reducing unnecessary data transmission
40 and power consumption. The methods and systems are versatile, applicable to various fields such as secure access control, industrial automation, and smart home environments, with further applicability not limited to the systems described herein.

[0126] Figure 5 illustrates a system architecture diagram for pairing initiation. Specifically, Figure 1 illustrates an implementation of system 500, where system 500 is configured to pair one or more user devices with one or more transmission devices. The devices are paired securely using a bidirectional connection that allows for real-time data exchange and synchronisation.

[0127] System 500 is directed towards secure, fast, and preferably simultaneous communication for synchronisation of content, e.g. an output at a content transmission device, and content interaction, e.g. an output at a user interaction device. The content is preferably media content, such as visual media content, e.g. movies, television shows, images, or live streams, and/or audio media content, e.g. music and podcasts. However, system 500 is not limited to media content, and can be used for alternative content and content interaction synchronisation and communication, such as games, simulations, social and communication interfaces, forms, surveys, exams, machine instructions, automated vehicle data, or other content not disclosed herein.

[0128] To ensure security of the connection, authentication and pairing is performed between devices. Beneficially, the pairing of devices using a bidirectional connection also allows for fast, bidirectional communication. Additionally, pairing is only completed, i.e. a handshake performed, after device communication is authenticated. Once paired, the bidirectional connection allows for improved synchronicity of content and content interaction between devices.

[0129] System 500 comprises: a user device 502, a transmission device 504, a user interface 506, a speaker 508, a content interface 510, a first request 512, a pairing request 514, a server 516, an identifier 518, a first sound code 520, and an identification code 522. Optionally, the user device 502 comprises or runs a user application, and the skilled person would understand that all reference to the methods, processes, and steps performed by the user device 502 are performed or initiated via the user application. Alternatively, or additionally, the transmission device 504 comprises or runs a transmission application, and the skilled person would understand that all reference to the methods, processes, and steps performed by the transmission device 504 are performed or initiated via the transmission application.

[0130] System 500 pairs user device 502 with transmission device 504 for output synchronisation between the user interface 506 and the content interface 510. User device 502 scans for content for user-content interaction via the user device 502 by broadcasting a first request 512. The first request 512 is then received by transmission device 504, which initiates the emission of a first sound code 520 from a speaker 508 associated with the transmission device 504. Using a microphone, the user device 502 detects the first sound code 520 and converts the first sound code 520 into an identification code 522. The identification code 522 is transmitted to the transmission device 504 for authentication and, if authenticated, a handshake is performed. A bidirectional connection (not depicted) is established between the user device 502 and the transmission device 504, allowing or real-time data exchange between devices.

[0131] Alternatively, or additionally, the transmission device 504 receives a first request 512 and transmits a pairing request 514 associated with the first request 512 to server 516. In response to receiving the pairing request 514, the server transmits an identifier 518 to the

transmission device 504. The identifier 518 is converted, e.g. encoded, into the first sound code 520 that is then emitted from speaker 508. The transmission device 504 receives an identification code 522, e.g. decoded from the first sound code 520 at the user device 502, from the user device 502. The identification code 522 is authenticated using the identifier 518 and, if authenticated, a handshake is initiated between the user device 502 and the transmission device 504.

[0132] Scanning for content comprises broadcasting a first request using a bidirectional communication protocol from the user device 502. The user device 502 is any device configured for user interaction with content, such as a mobile device, a tablet device, a portable computing device, wearable technology, or other devices comprising a microphone, a user interface, and bidirectional communication protocol capabilities. Optionally, as stated above, the user device 502 comprises or runs a user application, e.g. an app or application that performs or initiates any of the functionality described in relation to the user device 502. For example, a user application running on the user device 502 initiates the scan for content, resulting in the user device 502 broadcasting a first request 512.

[0133] The user interface 506 is configured for user interactions with the user device 502 including navigation, user input, user selection, and/or execution of user commands. In the example where the user device 502 is a mobile device, the user interface 506 is a screen such as a touch screen or display screen configured for user interaction, and output is displayed on the user interface 506. Examples of user interaction with the user interface 506 comprise any of: touching or viewing an output displayed on a touch screen; clicking, e.g. using a pointer, on buttons, links, or other interactive elements; and/or instructing the user device 502 to perform an action relating to an output on the user interface 506 via voice commands or motions, e.g. detected using the microphone or motion-detecting sensors, such as a camera or infrared sensor.

[0134] The bidirectional communication protocol comprises technologies and standards that enable the user device 502 to communicate with a transmission device 504, such as WebSockets. Beneficially, a WebSockets-based communication protocol provides full-duplex channels over a single, long-lived TCP connection. Unlike traditional HTTP, where the user device 502 continuously polls the transmission device 504 for updates or vice versa, WebSockets allows for real-time and near-real-time communication between the client and server. Therefore, WebSockets is especially useful in scenarios involving real-time media control or interactive content where low-latency communication is beneficial. For example, a user application running on the user device 502 sends commands, e.g. play, pause, volume up/down, to a paired transmission device 504, and the transmission device 504 sends updates to the paired user device 502, e.g. current playback position, a timestamp, buffer status, using WebSockets. Optionally, the communication from user device 502 and communication from the transmission device 504 is simultaneous. Not only is latency or lag minimised when using WebSockets, but the connection is more stable and secure than with legacy communication protocols.

[0135] Alternatives to WebSockets include: HTTP/2; Extensible Messaging and Presence Protocol (XMPP); Advances Messaging Queuing Protocol (AMQP) Message Queuing Telemetry Transport (MQTT); such as in situations where bandwidth is limited; Real-Time

Streaming Protocol (RTSP) or Real-Time Transport Protocol (RTP); Remote Procedure Call (gRPC); Web Real-Time Communication (WebRTC); Constrained Application Protocol (CoAP); Session Initiation Protocol (SIP); Bluetooth Low Energy (BLE) with Generic Attribute Profile); a different protocol or technology that facilitates bidirectional communication between user devices and media/transmission devices, preferably enabling real-time control, feedback, and interaction; or a combination thereof.

[0136] The transmission device 504 is a media device or a device capable of displaying or otherwise playing, transmitting, or emitting content, such as media content. For example, the transmission device is any of: a television device running a transmission application, e.g. Netflix™; a computing device; a media broadcast device; a micro-console such as a Chromecast™ or Apple TV™; or a device not listed here. The transmission device is associated with a content interface 510 such as a display screen and a speaker 508 preferably configured to emit sound frequencies detectable by a microphone of the user device 502 but less detectable to a user, such as ultrasound or near-ultrasound, e.g. audio tags generated by workstation 104. Optionally, the transmission device 504 comprises and/or runs a transmission application, e.g. an application configured to initiate or otherwise perform the functionality described in relation to the transmission device 504.

[0137] The transmission device 504 is configured to transmit content, e.g. via a transmission application, and respond to a user device 502 scanning for content. When the user device 502 scans for content, the transmission device 504 receives, or otherwise detects, a first request 512 broadcast from the user device 502. In the example where the first request 512 is broadcast using WebSockets, the transmission device 504 has WebSocket access or functionality, e.g. the transmission device 504 runs the transmission application that enables WebSocket access. Preferably, the bidirectional communication protocol used by the user device 502 and the bidirectional communication protocol used by the transmission device 504 is configured for secure communication, e.g. via the authentication and pairing steps described herein.

[0138] Beneficially, communication is further secured by use of sound codes such as the first sound code 520. The transmission device 504 is configured to convert the identifier 518 into the first sound code 520, e.g. using a predetermined encoding method to transform alphanumeric data of the identifier 518 into sound. Frequency range of first sound code 520 and any additional sound code is based on any one or combination of: detectability at a microphone, e.g. detectable by industry-standard microphones associated with user devices; detectability by the user, e.g. less detectable by a user; operational range of speakers, e.g. accuracy and ability for industry-standard speakers to emit the first sound code; or limiting interference, e.g. avoiding sound frequencies associated with content, conversations of a user, traffic, or any other standard noise source that could impact detectability and subsequent decoding of the first sound code 520. Ultrasound or near-ultrasound is preferable.

[0139] The user device 502 is configured to convert the first sound code 520 into an identification code 522, e.g. using a predetermined decoding method to transform the first sound code 520 into the identification code 522 or otherwise extract the identification code 522 from the first sound code 520. Optional predetermined methods include Dual-Tone Multi-Frequency (DTMF), morse code, and/or, preferably, Frequency-Shift Keying (FSK).

[0140] FSK is a modulation scheme where digital data, e.g. alphanumeric characters, is encoded by varying the frequency of a carrier wave. Different frequencies represent different binary values (e.g., 0s and 1s). These binary values can then be mapped to alphanumeric characters using a predefined encoding scheme such as American Standard Code for Information Interchange (ASCII). The FSK demodulator analyses the frequency changes in the received audio signal, converting the frequencies back into binary data. This binary data is then translated back into alphanumeric characters.

[0141] The user device 502 is further configured to interact directly with server 516. For example, output from the user device is based on information or data retrieved or otherwise received from the server 516. Preferably, the output is associated with content transmitted from the transmission device 504, received from server 516, and synchronised to the content using the methods and systems disclosed herein.

[0142] Identification and complex interaction with the content is sought by both users and content providers, requiring information that extends beyond basic details to include real-time data, contextual insights, and interactive elements like product details and scene-specific facts. User device 502 becomes more efficient as human-machine interactions are reduced, and the efficiency of communication and content providing systems is also increased. Moreover, reduced human-machine interactions and improved accuracy and reliability result in a safer more secure data retrieval system, as not only is the retrieved information relevant and cybersecure, but risks of malware or malicious content retrieval is effectively eliminated. Optionally, the first sound code 520 is used by user device 502 for identification of the output of the transmission device 504.

[0143] For example, the user device 502 performs a method for media content identification and content information retrieval using first sound code 520. Initially, the user device's microphone monitors a portion of an audio track emitted from a speaker 508, where the audio track is associated with specific media content. The user device 502 then detects whether an audio tag is embedded within this portion of the audio track, e.g. whether first sound code 520 is embedded. These audio tags are embedded at frequencies that are detectable by the microphone but are less noticeable to the user. If an audio tag is detected, the user device 502 extracts identification data from the audio tag. This identification data is linked to a known media content at a specific timepoint, e.g. the first sound code 520 comprises identification code 522 and a message timestamp. Optionally, if no audio tag is detected, the system resorts to an alternative method: it receives identification data based on the best match for an audio fingerprint from a database of multiple audio fingerprints via server 516. These fingerprints consist of one or more audio features extracted from the monitored portion of the audio track. Once the identification data is obtained, the user device 502 identifies the media content based on the known media content associated with the identification data. Subsequently, the user device 502 retrieves content information linked to this identification data.

[0144] Retrieving content information involves transmitting an information request, which includes the identification data, to a server device, e.g. server 516. The server 516 then responds with information related to one or more subjects relevant to the known media content at the specified timepoint. Each subject is linked to the known timepoint, ensuring that the retrieved information is contextually accurate and relevant to the media content being

identified. This method provides a seamless and efficient way for user devices to identify media content and access related information in real-time.

[0145] The above-described optional embodiment improves identification of media content and data acquisition, resulting in increases to efficiency, security, and reliability of data retrieval over known methods. For example, a user can identify the exact frame of a movie playing on TV and access related information, including actors, locations, and products featured in the scene, such as clothing or decor items. Multiple interactions between a viewer/user and a user device 502 are no longer required, e.g. as were previously needed for searching for the correct movie, searching for the correct scene, and then further searching for the related information. Such a reduction not only improves human-machine interaction, with more accurate, time efficient and secure data retrieval, but also presents an improvement to energy efficiency and battery life of the user device 502. When combined with synchronisation performed using a bidirectional connection as disclosed, efficiency, speed, system security, and user experience is further enhanced.

[0146] Only devices and/or apps configured with conversion, e.g. relevant encoding/decoding, functionality can pair, preventing unauthorised or malicious devices from compromising the communications, devices, or apps. For example, transmission device 504 transmits a pairing request 514 to the server 516 for device/user authorisation. Device/user authorisation optionally comprises any of: identification of the user device 502; identification of the user of the user device 502; identification of the transmission device 504; identification a user of the transmission device 504; or a combination thereof. The pairing request 514 comprises data associated with the first request 502, such as transmission ID, user ID, geo-location, or other identifying information. By identifying the pairing request 514, server 516 determines whether the pairing request 514 is associated with one or more trusted devices and/or users. Server 516 either generates an identifier 518 in response to identifying, and therefore authenticating, the pairing request 514, or extracts the identifier from a database, as shown in Figure 8.

[0147] In a further example, a software development kit (SDK) is used to provide a framework for allowing transmission devices and user devices to pair, e.g. SDK 1100 described below. The SDK for secure bidirectional communication, e.g. WebSocket communication, comprises components and/or functionality necessary for devices to perform the processes and methods described herein, enhancing the security of interactions between authorised devices. The transmission device 504 sends the first request 512 to the SDK and/or a transmission application utilising the SDK, embedded with the SDK, or otherwise associated with the SDK. The SDK generates a pairing request 514 and communicates the pairing request 514 to the server 516. The server 516 identifies the pairing request and sends an identifier 518 to the SDK. This allows only safe/secure interactions to and from the server 516, helping isolate the server from malicious devices or attacks.

[0148] Further optionally, secure communication is further enhanced using any of: encrypted communications, e.g. identifier 518, identification code 522 and/or bidirectional communications are encrypted using, for example, Transport Layer Security (TLS); firewall integration, e.g. where only authorised pairing requests 518 are received by the server 516; or IP filtering, where only requests from known IP addresses are allowed to establish bidirectional

connections. Following on from the SDK example, the SDK components and mechanisms ensure that only authorized devices can communicate over the bidirectional connection.

5 [0149] Beneficially, use of an SDK or associated transmission application allows for a variety of smart and non-smart transmission devices to operate the methods and processes described herein. For example, an internet-connected television, monitor, or screen can securely communicate using a bidirectional communication protocol and establish a bidirectional connection with a user device via the SDK or a transmission application, even though such functionality may not otherwise be securely implemented.

10 [0150] In another example, system 500 comprises a plurality of devices. For example, there is a plurality of user devices 502 comprising a first user device and a second user device, and a transmission device 504. Optionally, both the first user device and the second user device independently pair and synchronise with an output of the transmission device 504, such as displayed media content. An output of the first user device, such as a first information associated with the displayed media content, is synchronised to update and adapt based on
15 the output of the transmission device 504. Similarly, an output of the second user device, such as a second information associated with the displayed media content, is also synchronised to update and adapt based on the output of the transmission device 504. The first information and the second information comprise either the same or different information, e.g. based on a user selection, data associated with a user of either user device, or another factor impacting the first
20 information relative to the second information. Preferably, both the first information and second information are associated with the output of the transmission device 504.

[0151] In another scenario, there is a plurality of transmission devices 504 and a user device 502. The user device 502 scans for content and receives a plurality of sound codes in response, e.g. one from each transmission device 504. The user device 502 pairs with a selected
25 transmission device 504 based on, for example, a user selection or a selection protocol. A preferable selection protocol depends on any of: a proximity threshold, e.g. measured using geo-location or sound code volume; data associated with historical user selection, user preference, or user controls; or the priority associated with the output of the transmission device. For example, if a beneficial alert associated with danger to a user was output from a
30 transmission device 504, even though music is playing from a different device, and consequently the user device 502 pairs with the transmission device 504 and updated information associated with the danger is output at the user device 502.

[0152] Optionally, system 500 is a system for pairing and synchronising a user device, e.g. a mobile device, with a transmission device using a bidirectional communication protocol, e.g. WebSockets, and sound, e.g. ultrasound. System 500 comprises a transmission device and a mobile device. The transmission device is configured to: receive pairing requests from the mobile device via WebSocket; send pairing requests to a server and receive an identifier code from the server; convert the identifier code from text to an ultrasound or near-ultrasound code; send the ultrasound or near-ultrasound code to the mobile device; receive a text code from the
35 mobile device via WebSocket; perform a handshake with the mobile device if the text code matches; and periodically send ultrasound or near-ultrasound signals to ensure the mobile device remains in the vicinity of the transmission device. The mobile device is configured to: initiate a scan for content using WebSockets; receive an ultrasound or near-ultrasound code
40

via a microphone; convert the received ultrasound or near-ultrasound code to a text code; send the text code via WebSocket to the transmission device; perform a handshake with the transmission device if the text code matches; synchronise with the transmission device via WebSockets; and periodically receive ultrasound or near-ultrasound signals to ensure the mobile device remains in the vicinity of the transmission device, and prompt the user to confirm presence if a signal is missed.

[0153] Figure 6 illustrates a system architecture diagram for synchronising output. Specifically, Figure 6 illustrates system 600, where system 600 is configured to synchronise content across paired devices. For example, system 600 comprises system 500, allowing paired devices to synchronise content.

[0154] System 600 comprises: a user device 502, a transmission device 504, a user interface 506, a speaker 508, a content interface 510, a first sound code 520, a first timestamp 602, a second timestamp 604, a first output 606, and a second output 608.

[0155] According to one embodiment, user device 502 synchronises a first output 606 at the user interface 506 with a second output 608 at the content interface 510 of the transmission device 504. For example, the user device 502 identifies a first timestamp 602 associated with the first output 606, e.g. by determining the current timestamp of the first output 606. The user device 502 then identifies a second timestamp associated with the second output, e.g. by extracting a timestamp encoded in the first sound code 520. The second timestamp 604 corresponds to a timestamp of the second output at the point when the first sound code 520 was generated and/or emitted. Alternatively, the second timestamp 604 is received from the transmission device 504 using the bidirectional connection. The difference between the first timestamp 602 and the second timestamp 604 is then reduced, thereby improving synchronicity between the first output 606 and the second output 608. If the first timestamp 602 is ahead of the second timestamp 604, the first output 606 is delayed, e.g. slowed down, to temporally correspond to the second output 608. If the first timestamp 602 is behind the second timestamp 604, the first output 606 is advanced, e.g. sped up, to temporally correspond to the second output 608.

[0156] According to another embodiment, transmission device 504 synchronises the second output 608 with the first output 606. The transmission device 504 receives the first timestamp 606 from the user device 502 using the bidirectional connection, and identifies a second timestamp 608. The difference between the first timestamp 602 and the second timestamp 604 is then reduced, thereby improving synchronicity between the first output 604 and the second output 608. If the first timestamp 602 is ahead of the second timestamp 604, the second output 608 is advanced, e.g. sped up, to temporally correspond to the first output 606. If the first timestamp 602 is behind the second timestamp 604, the first output 606 is delayed, e.g. slowed down, to temporally correspond to the second output 608.

[0157] Optionally, the user device 502 is further configured for sound-based content synchronisation. Using the microphone, the user device 502 monitors a portion of an audio track emitted from a speaker to detect embedded audio tags. Audio tags encode, or are otherwise associated with, the second timestamp 603. These tags are embedded at frequencies detectable by the microphone but less noticeable to a user, e.g. using the same configuration as for first sound 520. Upon detecting an audio tag, user device 502 extracts

identification data associated with known media content for the second timestamp 604. If no audio tag is detected, the user device 502, via a server e.g. server 516, uses audio fingerprinting to identify the second timestamp 604 by matching audio features against a database of known fingerprints. This dual approach ensures accurate identification of timestamps, even in the absence of embedded tags. Once the second timestamp 604 is identified, the user device 502 alters first output 606 so the first timestamp 602 matches the second timestamp 604.

[0158] Figure 7 illustrates a system architecture diagram for pairing maintenance. Specifically, Figure 7 illustrates system 700, where system 700 is configured to maintain, or terminate, the pairing of devices. For example, system 700 comprises system 500, where paired devices remain connected or close the connection as required.

[0159] System 700 comprises: a user device 502, a transmission device 504, a user interface 506, a speaker 508, a bidirectional connection 702, a second sound code 704, a maintenance request 706, a first maintenance user selection 708, a sustain instruction 710, and a termination instruction 712.

[0160] After the user device 502 is paired with the transmission device 504, a second sound code 704 is transmitted. The second sound code 704 is transmitted at a stochastic/randomised interval, at consistent/uniform interval, or at an otherwise predetermined interval after transmission of the first sound code 520. The second sound code 704 ensures the bidirectional connection and/or synchronised output between the user device 502 and the transmission device 504 is still required. If the second sound code 704, initiated by the transmission device 504, is detected at user device 502, the user device 502 is still within the vicinity of the transmission device 504. If the second sound code 704 is not detected at the user device 502, the user device 502 may no longer be in the vicinity.

[0161] If the second sound code 704 is detected within a predetermined interval at the user device 502, the bidirectional connection is sustained. Sustaining a bidirectional connection, such as a full-duplex connection established via WebSockets, optionally comprises one or more processes and mechanisms that ensure the connection remains stable, secure, and functional over time. Examples include, but are not limited to, the following: connection maintenance, i.e. keep-alive mechanisms, such as ping-pong frames or heartbeat messages; data framing and fragmentation; messaging handling and routing; error handling and recovery, such as detecting invalid frames or unexpected disconnections; flow control and congestion management; session state management and state update mechanisms; or other processes to sustain a stable and secure bidirectional connection.

[0162] The bidirectional communication is terminated or otherwise closed if; the second sound code 704 is not detected within a predetermined interval at the user device 502; a geo-location associated with the user device 502 exceeds a predetermined threshold, e.g. 10 meters; a user interaction with the user device 502 indicative of no longer requiring output synchronisation; a termination threshold is exceeded, e.g. an hour or less without a user interaction at the user device 502 or the transmission device 504; or a combination thereof. Terminating a bidirectional connection, such as a full-duplex connection established via WebSockets, optionally comprises graceful closure, e.g. the user device 502 sends a close frame to transmission device 504, or a forced closure, e.g. either the user device 502 or the transmission

device 504 closes the connection forcefully. Timeout mechanisms help prevent stale connections from consuming resources unnecessarily.

[0163] According to one embodiment, a second sound code 704 is emitted from speaker 508. If it is detected by the user device 502, the bidirectional connection 702 is sustained. If it is not detected at the user device 502, a maintenance request 706 is displayed on the user interface 506, such as a question "Are you still watching?". A first maintenance user selection 708 is associated with a user interaction in response to the maintenance request 706 indicative of sustaining or terminating the bidirectional connection. If the first maintenance user selection 708 is indicative of sustaining the bidirectional connection, a sustain instruction 710 initiates maintenance of the bidirectional connection 702 and, optionally, any time-related thresholds for determining whether to sustain or terminate the bidirectional connection 702 are reset. If the first maintenance user selection 708 is indicative of terminating the bidirectional connection, a termination instruction 712 initiates termination of the bidirectional connection 702. Optionally, a second request 714 is broadcast from the user device 502 to scan for updated content.

[0164] Second sound code 704 is optionally first sound code 520 repeated at predetermined intervals. Alternatively, second sound code 520 is a separate sound code associated with updated data. For example, the first sound code 520 optionally comprises a first timestamp associated with the output of the transmission device 504 at the time the first sound code 520 was generated. The second sound code 704 comprises a second timestamp associated with the output of the transmission device 504 at the time the second sound code 704 was generated. If the output has been stationary, paused, or otherwise chronologically unchanged, the second timestamp is unchanged relative to the first timestamp, and the second sound code 704 is optionally first sound code 520. If the output has changed, e.g. advanced or reversed chronologically, the second timestamp is different to the first timestamp, and second sound code 704 is different to first sound code 704. In another example, second sound code 704 comprises data associated with the pairing between user device 502 and transmission device 704.

[0165] Figure 8 illustrates a system architecture diagram for identification retrieval. Specifically, Figure 8 illustrates system 800, where system 800 is configured to identify requests for pairing authentication. For example, system 800 comprises system 500, allowing associated devices to pair securely.

[0166] System 800 comprises: a transmission device 504, a first request 512, first request data 512-A, a pairing request 514, a server 516, an identifier 518, user device identifier 518-A, a database 802, a trusted file 804, and a transmission application 806.

[0167] As discussed above in relation to systems 500, 600 and 700, the server 516 receives a pairing request 514 comprising data associated with the first request 512, e.g. first request data 512-A, such as geographical location, timing data, device identifiers, application identifiers, or user identifiers associated with a device and/or application. First request data 512-A is optionally used by the server to search a database 802. Database 802 stores trusted files 804, e.g. a whitelist, associated with trusted and/or authorised devices and/or users. The identifier associated with user device 502, user device identifier 518-A, is then retrieved from

the associated trusted file 804. Alternatively, the server generates the user device identifier 518-A. Optionally, the server 516 comprises database 802.

5 [0168] Further optionally, the server receives the pairing request 514 and generates the identifier 518 after determining that the pairing request 514 meets predetermined safety protocols. A database 802 is optionally not required. Example safety protocols include: comparing first request data 512-A to a whitelist, determining whether first request data 512-A exceeds a time- or location-related threshold, or counting how many pairing requests 514 have been received within a predetermined timeframe.

10 [0169] Preferably, the server 516 communicates with a transmission application 806 running on the transmission device 504. The transmission application 806 is a content-embedded or service-embedded SDK preventing insecure communication between devices and the server 516. For example, the SDK/transmission application 806 is configured to receive the first request 512, generate and transmit the pairing request 514 to the server 516, receive or retrieve the identifier 518 from the server 516, convert the identifier 518 into the first sound code 520 and optionally instruct or otherwise initiate the speaker 508 to emit the first sound code 520, as described in relation to Figure 11.

20 [0170] The skilled person will understand that the transmission application 806 is applicable to systems 500, 600, and 700, where any or all functionality of the transmission device 504 is performed by the transmission application 806. Similarly, any or all functionality of the user device is optionally performed by a user application (not depicted here).

25 [0171] Figure 9 illustrates a user interface sequence for initiating content interaction. Specifically, Figure 9 illustrates a sequence of potential interaction initiations performed at the user device 504 for searching for and/or interacting with content of a transmission device 504. Any user interface disclosed is an option of user interface 502 of system 500, system 600, or system 700.

30 [0172] The first user interface sequence 900 comprises: a display 902; search user interface 506-A comprising an initiation user selection 904 and an application user selection 906; confirmation user interface 506-B comprising a confirmation user selection 908 and a reinitiation user selection 910; and paired user interface 506-C comprising a first paired content 912 and a content interaction user selection 914.

[0173] Optionally, user interface 506 comprises display 902. Alternatively, display 902 is optional and functionality of display 902 described herein is performed by the user interface 506.

35 [0174] First user interface sequence 900 is associated with a user of user device 502 searching for content using search user interface 506-A (e.g. initiating step 802 of method 800 below), selecting a desired content from one or more detected contents using confirmation user interface 506-B (e.g. resulting from step 804 and step 806, and/or initiating step 808 of method 800), and interacting with output synchronised to the selected content using paired user interface 506-C (e.g. resulting from step 810 and step 812 of method 800).

40 [0175] Search user interface 506-A comprises an initiation user selection 904. An interaction between a user of the user interface 506-A and the initiation user selection 904 is indicative of

user confirmation for user device 502 to scan for content, e.g. initiate method 800. Alternatively, a user of the user interface 506-A can choose not to initiate a scan for content by not interacting with initiation user selection 904, e.g. by interacting with application user selection 906 to initiate alternative functionality of the user device 502 or a user application running on the user device 502.

[0176] After a scan for content is initiated, confirmation user interface 506-B provides one or more confirmation user selections 908. The confirmation user selection 908 is associated with an identification code 522 extracted from a first sound code 520, and each confirmation user selection 908 is associated with a different identification code and a different sound code. The interaction between a user of user interface 506-B and confirmation user selection 908 is indicative of a user confirmation to pair with the associated transmission device 504 and/or synchronise output of the user interface 506 with the associated content, e.g. output 608. Optionally, the user of user interface 506-B interacts with reinitiation user selection 910 to initiate scanning for content. For example, reinitiation user selection 910 is initiation user selection 904.

[0177] After an associated content and/or transmission device 504 is confirmed, the user device 502 and transmission device 504 are paired, e.g. a handshake is performed, and a bidirectional connection is established. Paired user interface 506-C comprises the first paired content 912 and, optionally, a content interaction user selection 914. The first paired content 912 and/or paired user interface 506-C is optionally output 606, e.g. synchronised to output 608. The first paired content 912 is associated with an output at the paired transmission device. The one or more content interaction user selections 914 are associated with the first paired content 912, where user interaction with content interaction user selections 914 is indicative of a user interaction with the first paired content 912 and/or content output from the paired transmission device, e.g. output 608.

[0178] Example methods for a user interacting with any user interface described herein include touching a portion of a touchscreen, utilising voice commands, clicking on a button or portion of a displayed output using a pointer, or utilising other user-machine interaction methods.

[0179] Figure 10 illustrates a user interface sequence for terminating content interaction. Specifically, Figure 10 illustrates a sequence of potential interaction initiations performed at the user device 504 for sustaining/maintaining or terminating/closing a bidirectional connection between the user device 504 and a transmission device 506. Termination user interface 506-D is an option of user interface 506 of system 500, system 600, system 700, or first user interface sequence 900.

[0180] The second user interface sequence 1000 comprises: an optional display 902; paired user interface 506-C comprising a first paired content 912 and a content interaction user selection 914; termination user interface 506-D comprising a termination message 1002, a sustaining user selection 1004 and a terminating user selection 1006; and search user interface 506-A comprising an initiation user selection 904 and an application user selection 906.

[0181] Second user interface sequence 1000 is associated with a user of user device 502 interacting with content using paired user interface 506-C, terminating the synchronised

content interaction session using termination user interface 506-D, and optionally searching for updated content to interact with using search user interface 506-A.

5 [0182] Termination user interface 506-D comprises a termination message 1002, notifying the user that a condition for terminating the pairing session has been met. For example, the user device 502 may not have detected a second sound code 704 and/or a timing or location threshold has been exceeded. In another example, the bidirectional connection is compromised, insecure, or otherwise in need of termination and/or active maintenance, and termination message 1002 notifies the user.

10 [0183] Optionally, the termination message 1002 is associated with a termination time threshold. After a predetermined time interval measured from the generation and/or display of termination message 1002, the pairing session is closed, and the bidirectional connection is terminated. For example, no user interaction with the user interface 506-D is required for termination of the bidirectional connection. This timeout mechanism improves security of the bidirectional connection and improves efficiency by preserving computational and energy
15 resources.

[0184] Alternatively, or additionally, the termination message the termination message 1002 is associated with a request for user confirmation. A user interacting with sustaining user selection 1004 is indicative of the user confirming the pairing session is to continue, and the bidirectional connection is sustained. A user interacting with terminating user selection 1006
20 is indicative of the user confirming the pairing session is to close, and the bidirectional connection is terminated.

[0185] Figure 11 illustrates a system architecture diagram for an SDK. Specifically, Figure 11 illustrates SDK 1100, an SDK for secure communication and output synchronisation. For example, SDK 1100 is transmission application 806 and/or operates on transmission device
25 504, allowing transmission device 504 to pair securely with one or more user devices 502.

[0186] SDK 1100 comprises: communication module 1102, authentication module 1104, and interaction module 1106.

30 [0187] Communication module is configured to provide communication between the transmission device 504 and the user device 502 using a bidirectional communication protocol, such as WebSockets, and establish a bidirectional connection 702 between the transmission device 504 and the user device 502, e.g. a full-duplex connection. Communication module is further configured to communicate with the server 516. For example, communication module provides functionality for: receiving a first request 512 and an identification code 522 from the user device 502; transmitting a pairing request 514 and receiving an identifier 518 from the
35 server; and transmitting a first sound code 520 from speaker 508. Optionally, the communication module is configured to establish a secure WebSocket connection (WSS) using a TLS/SSL protocol.

[0188] Authentication module 1104 is configured to authenticate the user device 502, therefore improving security when establishing a bidirectional communication. For example,
40 authentication module 1104 provides functionality for: generating a pairing request 514, generating a first sound code 520, and comparing the identification code 522 to the identifier

518 to authenticate the user device 502 before the pairing session is initiated by establishing a bidirectional connection at the communication module 1102.

5 [0189] Optionally, generating the first sound code further comprises encoding the current timestamp using error-correcting, forward error correction, or Reed-Solomon. For example, the identifier is encoded in an acoustic signal, e.g. an ultrasound or near-ultrasound signal, utilising Reed-Solomon error correction coding, where data is transformed into a sequence of sound symbols with built-in redundancy to facilitate error detection and correction during transmission, emission, playback, or decoding.

10 [0190] Interaction module 1106 is configured to aid synchronisation of output from the transmission device 504, e.g. output 608 such as content provided by a content service provider, with output from the user device, e.g. output 606 such as paired user interface 506-C. For example, interaction module 1106 provides functionality for: determining and/or identifying a current timestamp associated with the output from the transmission device 504; identifying a message timestamp received from the user device 502 via the communication module 1102, optionally extracted at authentication module 1104; calculating a difference between the message timestamp and the current timestamp; and initiating adjustment of the output of the transmission device 504, e.g. speeding up or slowing down content, to minimise the difference; or initiating adjustment of the user device 502, e.g. instructing the user device 502 to speed up or slow down content, to minimise the difference.

20 [0191] According to one option, SDK 1100 further comprises a service module (not depicted) configured to embed a service such as a content providing service into SDK 1100. Additionally, or alternatively, SDK 1100 optionally further comprises an error correction module (not depicted) configured to detect and correct errors associated with communication module 1102, authentication module 1104, or interaction module 1106, such as an insecure bidirectional connection or errors in generating the first sound code, resulting in failed pairings.

[0192] According to another option, SDK 1100 further comprises a user interface module (not depicted) configured to display data associated with any module disclosed herein to a user of the transmission device 504 or a paired user device 502.

30 [0193] According to a further option, SDK 1100 further comprises a storage module (not depicted) further configured to store the data received via the communication module 1102 and/or processed by the authentication module 1104 or interaction module 1106.

35 [0194] Any of the above optional modules can be combined or otherwise integrated into SDK 1100. Additionally, any functionality provided by the SDK 1100 above may form a dedicated module or combined module. For example, SDK 1100 is directed to pairing and synchronising a transmission device with a user device, e.g. a mobile device, using bidirectional communication protocols, e.g. WebSockets, and sound, e.g. ultrasound. The SDK 1100 comprises: a module for receiving pairing requests from the mobile device via WebSocket; a module for sending pairing requests to a server and receiving an identifier code from the server; a module for converting the identifier code from text to an ultrasound or near-ultrasound code; a module for sending the ultrasound or near-ultrasound code to the mobile device; a module for receiving a text code from the mobile device via WebSocket; a module for performing a handshake with the mobile device if the text code matches; and a module for

periodically sending ultrasound or near-ultrasound signals to ensure the mobile device remains in the vicinity of the transmission device.

5 [0195] In another example, authentication module 1104 comprises a generation module configured to generate the pairing request 514 and the first sound code 520, and a comparison module, configured to compare the identification code 522 to the identifier 518. Optionally, any combination of modules described above comprise one or more additional modules, or are joined to form a combined module. For example, authentication module 1104 and interaction module 1106 form a single module.

10 [0196] The skilled person will understand that SDK 1100 can be utilised for transmission application 806 and is further applicable to systems 500, 600, and 700, where any or all functionality of the transmission device 504 is performed using SDK 1100.

15 [0197] In the context of the present disclosure, any of the several features described in Figures 1-4 and Figures 5-11 are optionally the same or serve similar functions. Specifically, the user device 502 is optionally the same as user device 106, providing a platform for user interaction with content. The server 516 is optionally the same as server 110, facilitating the storage and retrieval of content information and managing pairing requests. The speaker 508 is optionally the same as speaker 120, emitting audio tracks or sound codes for content identification and pairing. The database 802 is optionally the same as database 114, storing media content, related information, and trusted device data. The transmission device 504 is optionally the same as media content source 112, serving as a source of media content and facilitating content transmission. The identification code 522 is optionally the same as identification data 122, used for identifying media content and user devices. The first request 512 is optionally the same as request 124, initiating the process of content identification or pairing. The first output 606 is optionally the same as content information 108, providing information related to the media content. The user interface 506 is optionally the same as user interface 310-A, enabling user interactions with the device and content.

25 [0198] It is to be understood that the reference numbers used herein are not intended to be limiting. Elements of the features described in Figures 1-4 and Figures 5-11 are optionally shared based on the described functionality. This means that the skilled person would readily appreciate that combining elements from different figures is within the scope of the present disclosure. For example, the user device 502 from Figure 5 is optionally combined with the media interaction system 100 from Figure 1. Specifically, user device 502 can be used to interact with known media content 102, where the user device 502 detects audio track portion 118 emitted from speaker 508 (which is optionally speaker 120) and retrieves content information 108 from server 516 (which is optionally server 110). After, simultaneously or before content retrieval based on, e.g. system 100, the user device 502 synchronises with the transmission device 504/112, thereby improving efficiency and security of communication, data retrieval, and synchronisation between devices. In another example, the transmission device 504 from Figure 5 can be combined with the content synchronisation system 300 from Figure 3A. Specifically, transmission device 504 can emit audio signal 302 from speaker 508, where user device 310 (which is optionally user device 502) detects first audio tag 306 and second audio tag 308, and synchronises the first output 606 with the second output 608 based on the timestamps associated with the audio tags. These examples illustrate that the features

and elements described in Figures 1-4 and Figures 5-11 are interchangeable and can be combined in various ways to achieve the functionality described herein.

[0199] Figure 12 illustrates a flowchart of a method for performing media interaction. Specifically, method 1200 is a method performed by components of media interaction system 100 of Figure 1.

[0200] Method 1200 comprises: step 1202, processing a known media content at a workstation device; step 1204, interacting with a media content at a user device; and step 1206, transmitting content information to the user device.

[0201] Step 1202 comprises processing a known media content at the workstation device. The known media content is a media content that is already identified, or known, to the workstation device or the source of the media content, such that the known media content is associated with a unique content identifier. The content identifier is a name, code, or other form of ID that allows for the known media content to be identified or otherwise distinguished from other media content. Examples of known media content are movies, television shows, radio, podcasts, or other broadcast, streamed, or recorded media.

[0202] The workstation device is a computing system or platform capable of importing or otherwise obtaining the known media content and outputting or otherwise storing a modified media content. For example, the workstation device is a computer communicatively coupled to a server or a database, such as via an internet connection. In another example, the workstation device is workstation 104, configured to process known media content 102, of Figure 1. As described above in relation to Figure 1, the known media content comprises any media content associated with audio, such as television shows, movies or films, audiobooks, podcasts, music, video games, social media content, web series, streamed media or broadcast media. This allows for audio-based processing at the workstation device, audio-based identification at a server or user device, and audio-based synchronisation at the user device.

[0203] Step 1204 comprises interacting with a media content at the user device, such as a personal electronic device, emitting from a media playing device, such as a speaker. Interaction comprises identification of the media content and receiving information associated with the media content, e.g. from a server, a database, or the workstation device. If the media content being interacted with is a modified media content, e.g. the modified media content has been processed at step 1202, stored, and then accessed and played at a media playing device, the media content is identifiable at the user device. If the media content being interacted with is an unmodified media content, e.g. the media content has been processed at step 1202 but the original unmodified media content is being played at the media playing device, the media content is identifiable based on comparing the media content to stored known media contents, such as the known media content of step 1202. In instances where multiple known media contents are similar, e.g. the same audio track is used in multiple films, the user device is optionally configured to determine a best matching known media content, such as based on a user selection.

[0204] Step 1206 comprises transmitting content information to the user device. Either the server or workstation device transmits information relevant to the media content identified at step 1204 after the user device initiates a request for the content information. The content

information comprises subjects extracted from the media content, such as locations, people, services, items, objects, quotes, food and beverages, or other such identifiable content relevant to the media content being played, streamed, broadcast, or otherwise emitted. Content information comprises a visual depiction of an extract subject, subject text such as names or descriptions of the subject, links such as hyperlinks for further user-based interactions associated with the subject, or some combination thereof.

[0205] For example, the subject is a voting option relating to a televised contest being broadcast to a media playing device. The content information comprises a link to access a website hosting a poll, where a user of the user device can interact with the poll and vote. The user is not required to input data or interact repeatedly with the user device in order to reach the poll, and only the relevant poll is provided to the user. Not only does this increase efficiency regarding time and man-machine interaction, but less energy and computational resources are required, improving the battery life of the user device. There is also no opportunity for inaccuracy to be introduced into this information retrieval process, as there is no extensive search performed by the user and/or user device. Furthermore, security is increased, as the user device is not required to access multiple resources, e.g. multiple webpages, or interact repeatedly with a search engine or external database. This reduces the likelihood of malware being accessed by the user device. Additionally, false versions of the poll or other malicious content, which could otherwise appear to be related to the televised contest, are also not accessed. Therefore, the user device is less likely to be subjected to a phishing attack or other cybersecurity threat.

[0206] In another example, the media content is a streamed movie, and the subject is a location where the current scene of the streamed movie was filmed. Content information comprises information about the location, travelling to the location, hotels near the location, or other location-based data. The retrieval of such content information is optionally monitored, providing accessible metrics that can be analysed at the server, the workstation device, or output to an external system. Benefits include market research, trend analysis, demand forecasting, user experience optimization and improvement, traffic management, and environmental conservation. For example, conservation organizations can use location lookup data to identify areas of ecological importance or areas experiencing high levels of human activity. This information can inform conservation efforts and help mitigate environmental impact. Overall, monitoring location lookups can provide valuable insights across various domains, facilitating better decision-making and resource allocation.

[0207] In another example, the media content is an emergency weather radio broadcast and the subject is the name of a storm. Content information transmitted to the user device comprises safety information related to the impact of the storm in the location of the user device. In a further example, the media content is an emergency cybersecurity broadcast and content information comprises a patch to prevent a malicious virus from corrupting or otherwise negatively impacting the user device. In an additional example, the media content is a recording of a surgical procedure, the subject is a patient, and the content information comprises diagnostic and technical information associated with the surgical procedure, such as the heartrate and breathing rate of the patient, as the surgical procedure progresses. Optionally, the extraction of subjects and/or generation of content information is performed using machine learning models, allowing for near-real-time transmission of content information

synchronised to e.g. live broadcasts. These non-limiting examples result in targeted and efficient data dissemination, as well as accurate and efficient data retrieval.

5 [0208] Figure 13 illustrates a flowchart of a method for preparing content interaction at a workstation platform. Specifically, method 1300 is one method performed by components of media processing system 200 of Figure 2.

[0209] Method 1300 comprises: step 1302, import a media content to a workstation platform; step 1304, generate a timeline associated with the media content; step 1306, obtain one or more subject files; step 1308, assign each subject file to at least one timepoint; and step 1310, output an enriched media file.

10 [0210] Step 1302 comprises importing a media content to a workstation platform. The media content is associated with a plurality of subjects and a media content identifier. Importing the media content comprises obtaining the media content, e.g. from a server, database, or other media content source, downloading the media content or creating/generating the media content. The plurality of subjects comprises subjects relevant to the media content, as
15 described above in relation to method 1200, and the media content identifier is used to identify the media content from other known media content. Preferably, the media content identifier is unique for each imported media content.

[0211] For example, the workstation platform is media processing system 200 or, more specifically, workstation 202, the media content is known media content 204, and the media
20 content identifier is content identifier 218 of Figure 2. Once imported, the media content can be processed, e.g. as described in relation to step 1202 of method 1200. If the media content is to be processed, method 1300 proceeds to step 1304. If the media content is not to be processed or not to be processed completely, e.g. if a previous media content associated with the same media content identifier has previously been at least partially processed, method
25 1300 proceeds to any of steps 1304-610 to prevent unnecessary duplication and increase efficiency. In one example, the imported media content is already associated with a timeline (step 1304) and a plurality of subjects (step 1306). However, there are new subjects associated with the media content, resulting in steps 1306-610 being repeated as required.

[0212] Step 1304 comprises generating a timeline associated with the media content. The
30 timeline comprises a plurality of timepoints, where each timepoint is a predetermined time interval. Therefore, the timeline is generated by segmenting the imported media content into the predetermined time intervals. Timepoints on the timeline can then be used to synchronise content information associated with the media content to times when it is relevant, e.g. technical information about a vehicle is linked to a timepoint when the vehicle is present in the
35 media content.

[0213] For example, workstation 202 generates timeline 210 associated with media content 204 of Figure 2. If timeline 210 is already generated, e.g. if the media content 204 is at least partially processed previously, step 1304 comprises obtaining the timeline 210 and evaluating
40 timeline 210 to determine whether additional processing is required. Optionally, generating the timeline comprises determining the predetermined time interval for the plurality of timepoints. If a media content comprises relatively homogenous content, a longer timepoint results in fewer timepoints, less processing actions are required at workstation 202, and the resulting enriched

media file 228 is smaller, requiring less computational resources. However, if a media content comprises relatively inhomogeneous content over time, a shorter timepoint allows for more accurate synchronisation of content information associated with a subject to a time when the subject is relevant to the media content. The predetermined length of the timepoints may also impact the speed and/or the efficiency of media content identification.

[0214] Step 1306 comprises obtaining one or more subject files at the workstation platform. Each obtained subject file is associated with a subject of the plurality of subjects and comprises a visual depiction of the subject and a subject text. The subject file is obtained from: a database storing visual depictions and/or subject text for subjects, e.g. stored subject files associated with subjects of previously processed media content; the workstation, e.g. the subject file is generated using a machine learning model or from a knowledge base associated with the source of the media content; or an external source, e.g. information associated with a subject is searched for and retrieved at the workstation. Obtaining one or more subject files therefore optionally comprises generating a subject file associated with a subject of the plurality of subjects.

[0215] For example, the visual depiction is an image associated with the subject, such as a picture or video of the subject extracted from the media content or from a different source. The visual depiction is displayable on the user device, such as at a user interface and/or a display screen of the user device. The subject text is a name, description, or information associated with the subject, and is also displayable as textual data at the user device, such as at a user interface and/or a display screen of the user device.

[0216] Optionally, obtaining one or more subject files comprises using an artificial intelligence algorithm to: combine the visual depiction and the subject text; retrieve or generate the visual depiction or the subject text; or provide relevant data associated with the subject. For example, a generative algorithm system comprising one or more large generative models is used to generate subject text based on a visual depiction input or collate information about the subject e.g. for manual verification and/or editing.

[0217] Further optionally, step 1306 comprises editing, removing, adding to, or correcting the relevant data, the visual depiction, or the subject text. For example, if the one or more subject files are obtained from a database and are associated with a previous media content, information may be out of date or less relevant to the media content imported at step 1302. In another example, if the one or more subject files comprise data at least partially generated by artificial intelligence algorithms, information may be inaccurate or otherwise inappropriate in relation to the media content. Editing or other forms of correction or validation increase the relevance of the information associated with the subject files, which results in improved content interaction.

[0218] Step 1308 comprises assigning each subject file to at least one timepoint on the timeline. The subject file is linked to a timepoint where the subject is relevant to the media content, such as when the subject is shown, heard, or otherwise referenced within the media content. The subject file is optionally linked to a timepoint where the subject is not directly shown, heard, or otherwise referenced, but is indirectly relevant to the media content at that timepoint.

[0219] For example, if a watch is visible at a first timepoint and is not visible at a second timepoint but is still relevant, e.g. it is being spoken about in the media content or it will be visible again at a third timepoint, the subject file associated with the watch is assigned to the first timepoint and, optionally, the second timepoint. However, if the watch is not shown or otherwise relevant at a fourth timepoint, the subject file associated with the watch is not assigned to the fourth timepoint. This results in content information associated with the watch being displayed at a user device when the watch is relevant, and prevents display of irrelevant information at the user device when the watch is no longer relevant. Such targeted data retrieval reduces the amount of data being transmitted from a server and obtained at the user device, therefore reducing the memory and storage requirements of the user device in order to display relevant information.

[0220] Step 1310 comprises outputting an enriched media file, e.g. to a database, to a server, to a user device, or to a storage medium of the workstation device. The enriched media file comprises the timeline generated at step 1304, the one or more subject files associated with the timeline obtained at step 1306, and the media content identifier for linking the enriched media file to the media content obtained at step 1302. Optionally, outputting the enriched media file comprises storing the enriched media file in a database or otherwise downloading the enriched media file from the workstation platform. E.g. importing the media content to the workstation platform at step 1302 optionally comprises uploading a previously downloaded enriched media file output at step 1310. Therefore, one or more steps of method 1300 are repeated as necessary.

[0221] For example, the enriched media file is enriched media file 228 of Figure 2, comprising timeline 210, subject files 226, and content identifier 218. The enriched media file is accessible at the workstation device or a separate workstation platform, e.g. for further editing or modification, such as for editing the timeline or assigning additional or edited subject files to the timeline, or for optimising processing an additional media content that shares at least one subject of the plurality of subjects.

[0222] Optionally, the media content imported at step 1302 comprises an audio track. Method 1300 further comprises embedding an audio tag at a first timepoint within the audio track, where the audio tag comprises identification data associated with the media content identifier and the first timepoint. Embedding an audio tag generates an embedded audio track, e.g. step 1410 of method 1400. Further optionally, the enriched media file further comprises the embedded audio track, and any subject files assigned to the first timepoint at step 1308 are associated with the audio tag. This allows for audio-based identification and synchronisation of subject information retrieval. For example, at least one subject file assigned to the first timepoint is provided to a user device based on the audio tag being detected at the user device.

[0223] Further optionally, the media content imported at step 1302 is associated with a set of media contents. The set of media contents is associated with a group identifier unique to each set of media contents. For example, the set of media contents are individual episodes of a television series, the set of media contents are media content generated by the same media content creator, the set of media contents are media content sourced or owned by the same media content distributor, or the set of media contents are media content otherwise connected.

The enriched media file optionally further comprises the group identifier, making obtaining one or more subject files at step 1306 more efficient.

5 [0224] Figures 14A and 14B illustrate a flowchart of a method for subject extraction from media content at a workstation device. Specifically, method 1400 is another method performed by components of media processing system 200 of Figure 2 and/or audio recognition system 400 of Figure 4.

10 [0225] Method 1400 comprises: step 1402, obtaining a known media content comprising a known audio track; step 1404, generating a timeline for the known media content; step 1406, generating a first audio fingerprint; step 1408, linking identification data to the first audio fingerprint; step 1410, generating an embedded audio track associated with encoding identification data within the known audio track; step 1412, extracting one or more subjects from the known media content; step 1414, generating content information associated with each subject of the one or more subjects; and 1416, storing the first audio fingerprint, the embedded audio track, and the content information.

15 [0226] Step 1402 comprises obtaining a known media content, either at a server or at a workstation platform. The known media content comprises a known audio track, and is associated with one or more subjects relevant to the known media content. The known audio track comprises an audio signal, and is any form of audio, such as a sound recording, a song, speech, or other audio data that is detectable from a standard microphone, such as a
20 microphone of a personal computing device or mobile phone.

[0227] For example, step 1402 comprises step 1302 of method 1300 or is comprised within step 1202 of method 1200. Optionally, known media content is known media content 204 of Figure 2 or known media content 102 of Figure 1. E.g. the known media content comprises video content, interactive content, streamed content, broadcast content, user-generated
25 content, or another form of media. In another example, obtaining a known media content comprises importing the known media content from a server device, such as server 402 of Figure 4, or a media source, such as media content source 112 of Figure 1.

[0228] Step 1404 comprises generating a timeline for the known media content. The timeline comprises a plurality of timepoints, and each timepoint is a predetermined time interval such
30 that the timeline is segmented into predetermine time intervals. These segments can be used to synchronise subjects and associated content information to specific times within the known media content.

[0229] For example, step 1404 comprises step 1304 of method 1300 or is comprised within step 1202 of method 1200. Optionally, the timeline is timeline 210 of Figure 2. In another
35 example, the wherein a total duration of each timepoint, e.g. each predetermined time interval, is collectively at least a duration of the known media content. Alternatively, the known media content comprises a plurality of timelines., e.g. a timeline for each episode of a television series or a timeline for each known audio track where the known media content comprises a plurality of known audio tracks, such as when a movie is dubbed into different languages.

40 [0230] Step 1406 comprises extracting one or more audio features from the known audio track and using the one ore more audio features to generate an audio fingerprint. The audio fingerprint comprises the one or more audio features, where audio features are characteristics

of an audio signal, such as amplitude, frequency, timbre, duration, phase, envelope, harmonics, noise, dynamics, spatialisation, resonance, distortion, pitch, phase coherence, transients, or other characteristics that define and/or distinguish different audio signals.

5 [0231] For example, extracting one or more audio features comprises analysing an audio signal of the known audio track to identify or quantify selected audio characteristics. Analysing the audio signal comprises performing any of: a feature extraction technique comprising Fourier transforms, mel-frequency cepstral coefficients, chroma feature, spectral centroid, zero-crossing rate, root mean square energy, or other standard technique; statistical analysis, temporal analysis, feature scaling, or visualisation analysis; or one or more machine learning techniques, such as using a "Mel-Frequency Cepstral Coefficients" (MFCC) method.

[0232] MFCC is a feature extraction method widely used in speech and audio processing tasks, such as speech recognition, speaker identification, and music genre classification. For example, a typical MFCC method comprises pre-processing, mel-frequency conversion, cepstral analysis, feature selection, normalisation, and feature vector output. Pre-processing: 15 The audio signal is typically pre-processed by applying techniques such as windowing (e.g., using a Hamming window) and framing to divide it into small, overlapping segments. Mel-Frequency Conversion: The power spectrum of each frame is computed using techniques like the Fourier transform. Then, the power spectrum is transformed into the mel-frequency scale, which is a perceptually meaningful scale that better aligns with human auditory perception. 20 Cepstral Analysis: After converting to the mel-frequency scale, the log of the power spectrum is taken. This log-scaled spectrum is then transformed using the discrete cosine transform (DCT) to obtain the cepstral coefficients. These coefficients capture information about the spectral envelope of the audio signal. Feature Selection: Typically, not all cepstral coefficients are used for further analysis. Often, a subset of the coefficients is selected based on their 25 importance or relevance to the task at hand. Normalization: Optionally, the selected cepstral coefficients may be normalized to make them invariant to changes in overall volume or intensity. Feature Vector: The final output is a feature vector that represents the audio characteristics of the input signal. This feature vector can then be fed into a machine learning algorithm for tasks such as classification, clustering, or regression. MFCCs are effective for 30 capturing beneficial characteristics of audio signals, such as timbral information and spectral features, in a compact and discriminative manner. They have been successfully applied in various audio processing tasks due to their robustness and efficiency in representing audio data.

[0233] Generating an audio fingerprint by extracting one or more audio features from the 35 known audio track comprises generating a first audio fingerprint comprising one or more audio features extracted from the known audio track at the first timepoint. For example, one or more features are extracted from the known audio track at a first timepoint of the timeline, and these one or more features are used to generate the first audio fingerprint. As the known audio track in this example is inhomogeneous and varies from the first to a second timepoint, one or more 40 audio features extracted from the known audio track at the second timepoint will be different to the one or more audio features extracted at the first timepoint. A second audio fingerprint generated using the one or more audio features extracted at the second timepoint is therefore distinguishable from the first audio fingerprint. Both the known audio track and the first timepoint is identifiable based on the first audio fingerprint, and the known audio track and the

second timepoint is identifiable based on the second audio fingerprint. In other words, the first audio fingerprint is used to identify a known audio track, and the associated known media content, at a specific timepoint, as described above.

5 [0234] For example, the one or more audio features extracted from the known audio track are hashed or otherwise transformed into a coded identifier that forms the audio fingerprint. This coded identifier will change depending on the one or more audio features extracted from the known audio track, and can be compared with other coded identifiers to determine an audio fingerprint comprising similar or identical extracted audio features. Encoding the one or more audio features, e.g. using hashing technology or other encryption/encoding technologies to
10 store information about the one or more audio features, as an audio fingerprint allows for similar or identical audio fingerprints to be identified associated with similar or identical audio tracks, even if the original audio tracks are at different volumes, subject to different levels of noise, played using different audio codec technology or different audio quality, or subjected to other differing impacting factors.

15 [0235] Hashing the one or more audio characteristics optionally involves combining the various features extracted from an audio signal into a single hash value. An example approach to hash multiple audio characteristics comprises feature extraction, normalisation, concatenation, hashing, optionally salting, and output. Feature Extraction: Extract one or more audio characteristics or features from the audio signal, e.g. step 1412. Characteristics optionally
20 include MFCCs (Mel-Frequency Cepstral Coefficients), spectral features (e.g., spectral centroid, spectral bandwidth), rhythmic features (e.g., tempo, beat), or other relevant features like zero-crossing rate, energy, or pitch. Normalization: Normalize the extracted features to ensure consistency in scale and range. Normalization techniques such as min-max scaling or z-score normalization can be applied to each feature independently or collectively.
25 Concatenation: Concatenate the normalized feature vectors into a single feature vector. This creates a unified representation of all the extracted characteristics. Hashing: Apply a hashing function to the concatenated feature vector to generate a fixed-length hash value. Common hashing algorithms such as SHA-1, SHA-256, or MD5 are optionally used for this purpose, although the skilled person would understand that other algorithms would be also usable.
30 Salting: Optionally, a salt value is incorporated into the hashing process to enhance security and prevent pre-computed hash attacks. A salt value is a random string that is added to the input before hashing. This increases the security of future identification and synchronisation. Output: The output of the hashing process is a hash value that uniquely represents the combination of one or more audio characteristics extracted from the audio signal. The resulting
35 hash value can be stored or compared efficiently to identify similar audio content without the need to store the original audio data. Additionally, the choice of hashing algorithm and parameters optionally varies depending on the specific requirements of the application, including considerations for security, efficiency, and collision resistance. For example, the choice of hashing algorithm and parameters is tailored to the media interaction system 100.

40 [0236] Step 1408 comprises linking identification data to the audio fingerprint. The identification data is associated with the known media content at the relevant timepoint. For example, identification data linked to the first audio fingerprint of step 1408 comprises a content identifier for the known media content and the first timepoint. The identification data therefore identifies both the known media content associated with the audio track and the relevant

timepoint on the timeline corresponding to the linked audio fingerprint, allowing for audio-based content identification and synchronisation.

[0237] For example, a song being broadcast to a television comprises one or more audio features. The song is monitored by a microphone of a mobile phone, where the mobile phone and the television are independent devices. The one or more features are extracted to form a television audio fingerprint. The television audio fingerprint can then be compared to the first audio fingerprint generated at step 1406. If the first audio fingerprint and the television audio fingerprint is the same, the identification data linked to the first audio fingerprint is used to identify the television audio fingerprint and hence the song. The identification data is optionally received by the mobile phone, allowing for identification of the song. Further optionally, the one or more subject files associated with the song at the first timepoint (see method 1300 above) are received at the user device for synchronised display of relevant content information.

[0238] Step 1410 comprises embedding an audio tag within the known audio track obtained at step 1402, thereby generating an embedded audio track. The audio tag comprises the identification data of step 1408, such that embedding the audio tag encodes the identification data within the known audio track. At the first timepoint of the timeline generated in step 1404, a first audio tag is embedded within the known audio track, and the first audio tag comprises both the content identifier for the known media content and the first timepoint, allowing for audio-based content identification and synchronisation.

[0239] Preferably, the audio tag is detectable by a microphone of a user device but less detectable by a user of the user device or workstation device. For example, the audio tag is embedded at frequencies that are inaudible to a human, e.g. using frequency-shift keying modulation at defined intervals, where the defined intervals are designed to be beyond human detection in standard environmental conditions.

[0240] Frequency-shift keying (FSK) modulation is a digital modulation technique used in telecommunications and digital communication systems, and involves modulating a carrier signal's frequency based on a digital input signal. A digital input signal consists of binary data, where each binary symbol (bit) represents a specific digital value, typically 0 or 1. A carrier signal with a fixed frequency f_c is generated. This carrier signal will be modulated in frequency according to the digital input signal. In FSK modulation, different frequencies are assigned to represent different digital values. Typically, one frequency (e.g., f_1) is used to represent a binary "1" and another frequency (e.g., f_2) is used to represent a binary "0". Based on the digital input signal, the carrier signal's frequency is modulated to switch between the assigned frequencies. When the input signal is at a logic high (e.g., binary 1), the carrier frequency is shifted to f_1 . Conversely, when the input signal is at a logic low (e.g., binary 0), the carrier frequency is shifted to f_2 . The modulated signal, which now carries the information encoded as frequency shifts, is transmitted over the communication channel. At the receiver end, the modulated signal is received and demodulated to recover the original digital input signal. This process involves detecting the carrier frequency shifts and mapping them back to the corresponding digital values. FSK modulation offers several advantages, including simplicity, resistance to noise, and efficient use of bandwidth. To overcome potential limitations such as susceptibility to frequency offset and non-linear distortions in the transmission channel, variants of FSK, such as continuous-phase frequency-shift keying (CPFSK) and minimum-shift keying (MSK), are

optionally used. Alternatives such as CPFSK and MSK techniques apply the basic principles of frequency modulation based on digital input signals.

[0241] Both the first audio tag of step 1410 and the first audio fingerprint of step 1406 and step 1408 can be used to identify the known audio track. The first audio tag of step 1410 allows for direct extraction of the identification data associated with the known media content at the first timepoint, which results in faster identification of the known media content. Additionally, it does not matter whether the known audio track is associated with multiple known media content, or multiple timepoints, or multiple known media contents at multiple timepoints. The encoded identification data comprises the content identifier of the relevant known media, which is different from other content identifiers irrespective of e.g. common or mutual audio tracks.

[0242] In order for the audio tag to be detected, the embedded audio track must be played and detectable, which is not always possible for e.g. old media content without redistributing the soundtrack or with low bitrate audio file formats. For example, streamed media content does not always have a stable or high bandwidth internet connection, which could result in emitted sound not comprising detectable embedded audio tags. Therefore, the audio fingerprint of steps 1406 and step 1408 provide a fall-back mechanism when audio tags are not detectable or detected. While the generation of an audio fingerprint and comparison to other audio fingerprints linked to identification data takes longer than extracting identification data directly from an embedded audio track, and potentially requires storage of multiple audio fingerprints for comparisons, it allows old media content and lower quality audio tracks to be identifiable, both in terms of what media content is being monitored (the content identifier) and where in the media content monitoring is happening (the timepoint).

[0243] Step 1412 comprises extracting one or more subjects from the known media content. For example, subjects are extracted and labelled or otherwise categorised at one or more timepoints in the timeline generated at step 1404. The one or more subjects are items, locations, people, services, or other objects relevant to the known media content at any of the one or more timepoints. For example, a subject extracted at a first timepoint is a singer of the known audio track associated with the known media content, an object being worn by an actor at the first timepoint within the known media content, a location where the known media content at the first timepoint was created, or an actor who is heard at the first timepoint.

[0244] Optionally, extraction is performed using one or more artificial intelligence algorithm, e.g. using of a machine learning model such as an artificial neural networks or computer vision models. For example, the computer vision model is trained on image data to identify and classify objects within the image. Objects are analysed with one or more object tracking algorithms to track how many consecutive frames or timepoints a given object is part of, which is then linked to the timeline. Additionally, identified objects are optionally linked to other identified objects identified at different timepoints, resulting in merging of the identical objects at different timepoints.

[0245] Step 1414 comprises collating information associated with each of the one or more subjects extracted at step 1412. The collated information is used to generate content information associated with the identification data, which can be provided to a user device based on the identification data. For example, step 1412 and step 1414 comprise step 1308 of method 1300, and the information associated with an extracted subject is the subject file 226

of Figure 2. The combination of subject files relevant to e.g. the first timepoint is the collated information associated with the one or more subjects extracted from the known media content at the first timepoint. Therefore, generated content information represents information relevant to the known media content at the first timepoint.

5 [0246] Optionally, step 1414 is performed using one or more artificial intelligence algorithms. For example, the object identified at step 1412 is enriched with metadata from e.g. a multimodal generative artificial intelligence algorithm, where an extracted image of the object (a visual depiction of a subject) is used as an input for e.g. more detailed identification or to retrieve information associated with the object. Sources of information associated with the object can
10 be controlled or otherwise monitored, resulting in safe and accurate information retrieval e.g. from known authoritative sources, or can be generated, e.g. using large language models or other generative algorithms. Objects, and the associated images and text, are optionally stored, allowing for future identification of existing objects and the training or updating of machine learning model.

15 [0247] Further optionally, verification at the server or at the workstation device of object identification and linked timepoints results in improved accuracy and relevance during extraction of the one or more subjects. Verification optionally comprises linking objects together, adding additional objects, modifying associated metadata, selecting the image of the object, or adding precision to identified objects. Beneficially, combining multiple artificial
20 intelligence algorithms, such as classifiers at step 1412 and generative models and step 1414, results in an automated system for extracting subjects and/or generating content information. Optionally, the artificial intelligence algorithms are hosted externally, hosted at the server and/or communicated with from the server to decrease the computational resources required by the workstation and increase efficiency.

25 [0248] Step 1416 comprises storing the audio fingerprint, the identification data, the embedded audio track, and the content information. For example, step 1416 comprises storing enriched media file 228 of Figure 2 at the server or at a database. The identification data is retrievable, e.g. by the server, workstation or user device, based on the audio fingerprint or the embedded audio track. This allows for audio-based media content identification and
30 synchronisation. The content information is retrievable based on the identification data, resulting in data retrieval of relevant information, both in terms of contextual relevance and temporal relevance.

[0249] Optionally, the content information associated with additional timepoints is also retrievable based on the identification data. For example, content information associated with
35 a first timepoint and a second timepoint is retrievable based on the identification data associated with the first timepoint, where the second timepoint is within an adjustable buffer period. The buffer period is adjustable at the workstation, server, or user device, and is based on a user selection, tailored to the known media content, or based on physical or computational restraints associated with the devices used in performing e.g. method 1400, such as based on
40 the speed of data communication between devices. Alternatively or additionally, the adjustable buffer period is extended or otherwise adjusted based on whether a lock-on indicator received, as detailed below in relation to method 1600.

[0250] In another example, the server will look up content information associated with timepoints within the adjustable buffer period and return all objects and the timepoints relevant to the objects in the adjustable buffer period. Optionally, such as based on a user request or selection for content information, the user device will receive a sequential collection of content information, e.g. visual depictions of subjects. Further audio-based monitoring of the known media content is optionally used to keep the sequential collection synchronised to the known media content. Further optionally, the user device receives and/or displays additional information, e.g. subject text or additional content information generated at step 1414, on a user interface of the user device, such as a display screen, based on a user interaction with the user device. In one example, interacting with the visual depiction of a subject, e.g. tapping or clicking on a visual depiction of the subject, results in a detailed view of the subject with varying information and possible future interactions based on the type of subject.

[0251] Optionally, step 1416 comprises storing the first audio fingerprint, the embedded audio track, and the content information at a database. The database is configured to retrieve identification data based on an audio fingerprint or the embedded audio track and/or retrieve content information based on the identification data. Alternatively, the database is configured for a server to retrieve the identification data or content information as described, where the server is connected to the database. Preferably, the workstation device performing steps of method 1400 communicates with the server configured to access the database. In another example, the workstation device communicates with the database directly, or the server performs the steps of method 1400.

[0252] Optionally, steps of method 1400 are repeated. For example, step 1406 is repeated for a second audio tag. The second audio tag comprises identification data associated with the known media content at a second timepoint, where the second timepoint is of the plurality of timepoints within the timeline generated at step 1404. Step 1410 comprises embedding the second audio tag within the known audio track, such that the embedded audio track comprises the first audio tag and the second audio tag.

[0253] Figure 15 illustrates a flowchart of a method for media content identification and content information retrieval at a user device. Specifically, method 1500 is one method performed by components of content synchronisation system 300 of Figures 3A and 3B.

[0254] Method 1500 comprises: step 1502, monitor a portion of an audio track associated with media content; step 1504, determine if there is an audio tag embedded within the portion; step 1506, extract identification data from the audio tag; step 1508, retrieve identification data based on one or more extracted audio features; step 1510, identify the media content based on the identification data; and step 1512, retrieve content information associated with the identification data.

[0255] Step 1502 comprising monitoring, using a microphone of the user device, a portion of an audio track emitting from a speaker, wherein the audio track is associated with a media content. The microphone is attached to, or communicatively coupled with, the user device, and is configured to collect audio signals. Monitoring a portion of an audio track comprises collecting an audio signal emitted from the speaker, where the speaker is attached to or communicatively coupled with a media playing device. Preferably, the media playing device and the user device are independent devices.

[0256] For example, the media content is a film being streamed on a television device, and the soundtrack of the film emitting from the television device speaker comprises the audio track. The microphone of a mobile device monitors a portion, e.g. a section or interval, of the audio track. The portion is any duration between a microsecond to the total duration of the audio track, and is preferably between 1 and 10 seconds, e.g. 2 seconds.

[0257] Step 1504 comprises detecting whether there is an audio tag embedded within the portion of the audio track. The audio tag is embedded at frequencies detectable by the microphone, e.g. frequencies detectable by a standard in-built microphone of a standard mobile or personal computational device. Preferably, the audio tag is embedded at frequencies less detectable by a user of the user device, e.g. at frequencies where a person would not be able to distinguish the audio tag from the audio track and/or surrounding noise, or at frequencies otherwise inaudible to a standard human.

[0258] Returning to the example of step 1502 above, an audio tag is detected by the microphone of the mobile phone device within the audio signal being monitored at step 1502. Optionally, the detected audio tag is embedded using frequency-shift keying modulation at defined intervals. These intervals are optionally tailored to the audio track, associated media content, or to meet physical or computational restraints associated with standard microphones, speakers, or human audio detection. As the audio tag is detected, method 1500 proceeds to step 1506. Otherwise, method 1500 proceeds to step 1508.

[0259] Optionally, step 1504 comprises application of one or more correction algorithms to compensate for noise, interference, or other audio degradation factors. This results in more accurate and efficient audio tag detection. Alternatively, or additionally, detecting whether there is an audio tag is limited to a predefined time frame, such as between 1 and 10 seconds, e.g. 3 seconds, or depending on an estimated time between audio tags embedded within the audio track, or depending on an estimated time between timepoints. For example, if an estimated two audio tags should have been detected during the predefined time frame and no audio tag is detected, method 1500 proceeds to step 1506.

[0260] Step 1506 comprises extracting identification data from the audio tag if the audio tag was detected at step 1504. The identification data is associated with a known media content at a known timepoint. For example, the identification data comprises a content identifier, e.g. content identifier 218 of Figure 2, and a timepoint, e.g. timepoint 212. The audio tag is therefore encoded identification data within an audio tag. An audio track with embedded audio tags, an embedded audio track, is the audio track enriched with machine-readable data comprising the content identifier and time in the audio track at given intervals, e.g. step 1410 of method 1400.

[0261] Preferably, the identification data is extracted from the audio tag at the user device. Beneficially, this results in quick and accurate identification and, optionally, synchronisation of media content at the user device. Alternatively, the audio tag is transmitted to e.g. a server for identification data extraction. If identification data is successfully extracted, method 1500 proceeds to step 1510. Otherwise, step 1506 is repeated or method 1500 proceeds to step 1508.

[0262] Step 1508 comprises retrieving the identification data based on a best match for an audio fingerprint from a plurality of audio fingerprints. The audio fingerprint comprises one or

more audio features extracted from the portion of the audio track. For example, step 1508 comprises extracting one or more audio features to generate an audio fingerprint, e.g. step 1406 of method 1400, such as by hashing the one or more audio features or otherwise encoding the one or more audio features.

5 [0263] The one or more audio features are extracted at the user device or at a server. If the one or more audio features are extracted at the user device, the audio features or the audio fingerprint comprising the one or more audio features is transmitted to the server. Extracting the one or more audio features at the user device increases efficiency. If the one or more audio features are not extracted at the user device, the portion of the audio track is transmitted to
10 the server, and the audio fingerprint is generated at the server. Extracting the one or more audio features at the server decreases computational resource requirements at the user device.

[0264] The generated audio fingerprint is compared to other audio fingerprints, such as a plurality of audio fingerprints stored in a database or at the server. The plurality of audio
15 fingerprints are associated with a plurality of portions of an audio track, such as an audio track at different timepoints, and/or portions of a plurality of audio tracks, such as audio tracks associated with one or more media contents that have been previously processed to generate audio fingerprints associated with the media contents. As each of the plurality of audio fingerprints is associated with identification data, identification data can be retrieved based on
20 each audio fingerprint. The best match is the stored audio fingerprint most similar to the generated audio fingerprint, e.g. it is the same audio fingerprint, and the identification data associated with the best match is retrieved by the server and subsequently received at the user device.

[0265] Step 1510 comprises identifying the media content based on the known media content
25 of the identification data. For example, the identification data of the best match is associated with a known media content, and therefore the media content of step 1402 is identified as the known media content. If an audio track is optionally associated with multiple known media contents, the best match of step 1508 is optionally a plurality of best matches, and step 1510 then comprises selecting the media content from a plurality of known media contents.
30 Returning to the above example, the audio track is a song played during the film on the television device. The song is also within the soundtrack of a different film that is not being played on the television device. Therefore, identification data associated with both the film on the television and the different film is received at the user device at step 1508. Step 1510 comprises requesting a selection from a user of the user device and receiving a selection from
35 the user, e.g. displaying the titles of both films based on the content identifier and a user tapping or clicking on the correct title.

[0266] Step 1510 optionally further comprises identifying a timepoint within the media content based on the identification data. Not only is the media content identified, but also a time position where the media content is currently, such as a minute and second corresponding to which
40 part of the media content is being played at e.g. the television device. For example, the film is identified as Film-A, and the timepoint is identified as 01:17:33 using hh:mm:ss formatting. In this example, the timepoint comprises a predetermined interval of one second, such that the subsequent timepoint would be identified as 01:17:34. If the timepoint comprised a

predetermined interval of five seconds, the timepoint would be identified as 01:17:30 – 01:17:35.

[0267] Optionally, the microphone is switched on and off to match the defined intervals which the audio tags are embedded within the audio track. Beneficially, audio tags are then present and detectable while the microphone is switched on, and the microphone is switched off when audio tags are not present and hence not detectable, reducing the energy requirements of the microphone and improving battery life of the user device without impacting synchronisation processes based on monitoring of the audio track. Further optionally, if no match is found, e.g. over a longer period of time than the defined interval due to environmental noise, the microphone is kept on until either an audio tag is detected or to perform step 1508.

[0268] Step 1512 comprises retrieving content information associated with the identification data. For example, retrieving the content information generated at step 1414 of method 1400 and/or subject files 226 of Figure 2. A request for content information is received, such as based on a user interaction with the user device, and the identification data is transmitted to the server from the user device. Optionally, content information is information associated with one or more subjects relevant to the known media at the known timepoint. Each subject of the one or more subjects is linked to the known timepoint, or surrounding timepoints within the adjustable buffer period.

[0269] If identification data was received from the server, e.g. step 1508 was performed to identify the media content, step 1512 is optionally performed with step 1508, such that the identification data and content information is received from the server without an additional request for content information being received. Alternatively, an additional request for content information comprises a user selection, e.g. if there is a plurality of best matches.

[0270] As the content information retrieved is associated with the identification data, which comprises the timepoint of the currently playing media content, the retrieved content information is synchronised with the playing of the identified media content. Returning to the above example, Film-A is playing on a television device and is at 01:17:33. There are a plurality of subjects associated with Film-A at 01:17:33, including an actor narrating, who is not visible at 01:17:33 but can be heard, a species of shark, which is visible at 01:17:33, a location where the shark was filmed, and a service for donating to a shark-preservation charity, which is mentioned at a different timepoint in Film-A. The content information comprises: information associated with the actor, such as their name and other known media content they have appeared in; information associated with the shark, such as the species name and conservation status; information associated with the location, such as the name and the cost of travel between the user device location and the filmed location; and information associated with the service, such as the name of the charity and a link to their website. Optionally, step 1512 comprises displaying content information on the user device based on the timepoint. The displayed content information is then relevant to the media content at the known content. For example,

[0271] Optionally, step 1512 further comprises transmitting an information request comprising the identification data, e.g. from the user device to the server or database. Information associated with the one or more subjects relevant to the known media at the known timepoint, as per the transmitted identification data, is then obtained from the server or database at the

user device. Each subject of the one or more subjects is linked to the known timepoint, e.g. step 1308 of method 1300. Further optionally, the information request is transmitted based on a user interaction with the user device. Alternatively, the information request is transmitted automatically upon identification of the media content at step 1510, or based on alternative
5 criteria linked to whether the user is interacting with, e.g. in proximity to, the identified media content.

[0272] Figures 16A and 16B illustrate a flowchart of a method for content interaction at a user platform. Specifically, method 1600 is another method performed by components of content synchronisation system 300 of Figures 3A and 3B. Any of the steps of method 1600 can be
10 combined with any of the steps of method 1500 for the identification of media content and/or synchronisation of retrieved content information.

[0273] Method 1600 comprises: step 1602, monitor an audio signal; step 1604, obtain a selected audio tag from the audio signal; step 1606, extract identification data from the selected audio tag; step 1608, determine if a lock-on indicator is received; step 1610, obtain a request
15 for a first subject file; step 1612, receive the first subject file; step 1614, display data from the first subject file; step 1616, reduce monitoring of the audio signal; step 1618, request a plurality of subject files; step 1620, receive a plurality of subject files; step 1622, display data associated with the first subject file; and step 1624, display data associated with a second subject file.

[0274] Step 1602 comprises monitoring an audio signal, where the audio signal comprises one
20 or more audio tags embedded within the audio signal. For example, the audio signal is from an embedded audio track generated at step 1410 of method 1400. In another example, step 1602 comprises steps 1502, 1504, and 1506 of method 1500. The audio signal is associated with an audio track of a media content, and is emitted from a speaker of a media playing device. Preferably, the media playing device is independent to a user device associated with
25 monitoring the audio signal. For example, the audio signal is collected using a microphone connected to the user device, and monitored at a user platform of the user device. Optionally, the user platform is the user device. Alternatively, the user platform is a software interface or application running on the user device.

[0275] Step 1604 comprises obtaining a selected audio tag from the audio signal. The selected
30 audio tag is used to identify a media content, and is optionally the first audio tag detected during monitoring of the audio signal. Alternatively, it could be a subsequent audio tag, such as when monitoring of the audio signal at step 1602 is repeated. The audio tag comprises encoded identification data associated with a specific media content, e.g. the media content comprising the audio signal of step 1602.

[0276] Optionally, obtaining a selected audio tag comprises extracting a plurality of audio tags
35 from one or more audio signals. For example, multiple media content is playing simultaneously, resulting in multiple audio tags being detected. The selected audio tag is determined from the plurality of audio tags based on a user interaction with the user platform, such as selecting the audio tag associated with a media content the user wishes to interact with.

[0277] Step 1606 comprises extracting identification data from the selected audio tag. The
40 identification data comprises a media content identifier, e.g. a content identifier unique to a specific media content, and a timepoint, such as a first timepoint, which is indicative of the

current time location within the media content at which the audio signal is being emitted and monitored. Step 1606 therefore comprises identifying the media content associated with the audio signal monitored at step 1602.

5 [0278] Alternatively, steps 1604 and 1606 comprise step 1508 of method 1500. For example, when an audio tag is not detectable, identification data is received at the user platform based on one or more audio features extracted from the audio signal.

10 [0279] Step 1608 comprises determining if a lock-on indicator is received at the user platform. The lock-on indicator is indicative of whether the user platform should synchronise with the media content identified at step 1606. For example, the lock-on indicator is a user interaction with a user device running the user platform, such as a tap on a touch screen display, or a criteria determined at the user platform, such as whether the user device is moving away from the media content, e.g. based on location data or monitoring of the audio signal at step 1602, or whether an audio tag from a different audio signal is detectable.

15 [0280] If a lock-on indicator is received, method 1600 proceeds to step 1616. If a lock-on indicator is not received, method 1600 proceeds to step 1610.

20 [0281] Step 1610 comprises obtaining a request indicator for a first subject file, where the request indicator is a user interaction with the user platform or another indication to proceed with retrieving information associated with the identification data. The first subject file is associated with the media content of the identification data at the timepoint of the identification data, such as the first timepoint. One example indication to proceed is an estimated proximity of a user to the media content, e.g. based on the volume of the audio signal, and another example is whether the user platform is active on a user device, e.g. the user platform is displayed on the user device.

25 [0282] Step 1612 comprises transmitting the identification data extracted or received at step 1606 to a server device, e.g. a server, and receiving, from the server device, the first subject file. The first subject file comprises a first visual depiction, e.g. an illustration of the first subject, and a first subject text, e.g. a name and/or textual description or information associated with the first subject. For example, the first subject file is subject file 226 of Figure 2. Optionally, step 1612 comprises receiving a plurality of first subject files, where each subject file is associated with a subject relevant to the media content at the e.g. first timepoint.

30 [0283] Step 1614 comprises initiating display of the first visual depiction or the first subject text. For example, step 1614 comprises displaying the first visual depiction and the first subject text alongside the first visual depiction. In another example, step 1614 comprises displaying the first visual depiction and, upon receiving a user interaction with the user platform, displaying the first subject text. In another example, initiating display comprises receiving a user interaction with the user platform indicative of whether to display e.g. the first visual depiction, the first subject text, or neither. Optionally, method 1600 proceeds to step 1602.

35 [0284] For example, repeating step 1602 comprises monitoring the audio signal, wherein the audio signal comprises a new audio tag. Step 1604 is repeated and a new audio tag is obtained from the audio signal. New identification data is extracted from the new audio tag at step 1606, where the new identification data is associated with the media content identifier and a second timepoint or a new media content identifier.

40

[0285] Optionally, each subject of the plurality of subjects is associated with a classification. Each subject of the plurality of subjects is associated with a visual depiction and a subject text, such that e.g. the first subject file comprises a plurality of visual depictions and a plurality of subject texts. Further optionally, initiating display at step 1614 comprises selecting e.g. a visual depiction of a subject from the plurality of subjects to display based on the classification.

[0286] Step 1616 comprises reducing monitoring of the audio signal. For example, the microphone is switched on and off to match the defined intervals where subsequent audio tags are embedded within the audio track, or the microphone is switched off for a longer interval with only periodic or sporadic detection of audio tags to validate synchronisation. Optionally, the microphone is switched off until an indicator initiates the microphone switching on to monitor an audio signal, such as a user interaction with the user platform or a change in user device location or activity.

[0287] Step 1618 comprises transmitting the identification data to the server device, thereby requesting a plurality of subject files from the server device. The plurality of subject files is associated with the media content of the identification data at a plurality of timepoints, such as the first timepoint and a predetermined number of additional timepoints, e.g. based on the adjustable buffer period of method 1500. The predetermined number of additional timepoints are optionally subsequent timepoints, such as a second timepoint and a third timepoint, e.g. when the media content is emitting typically. However, if monitoring of the audio signal indicates that the media content is being emitted at an increased rate, e.g. it is being fast-forwarded, the predetermined number of additional timepoints are optionally selected subsequent timepoints, such as a fourth timepoint and an eighth timepoint as to maintain synchronization between the user platform and the media content. Alternatively, if monitoring of the audio signal indicates that the media content is being paused, additional timepoints are optionally timepoints surrounding the first timepoint. Alternatively, if monitoring of the audio signal indicates that the media content is being rewound or played in reverse, additional timepoints are optionally timepoints preceding the first timepoint.

[0288] Step 1620 comprises receiving, from the server device, the plurality of subject files. For example, the server uses the identification data received at step 1618 to retrieve the plurality of subject files from the database, wherein each subject file is relevant to one or more subjects extracted from the media content, e.g. steps 1412 and 1414 of method 1400.

[0289] Step 1622 comprises initiating display of the first visual depiction or the first subject text, as described in relation to step 1614.

[0290] Step 1624 comprises initiating display of a second visual depiction or a second subject text, as described in relation to step 1614. The second visual depiction and second subject text are comprised within a second subject file of the plurality of subject files received at step 1620. Preferably, display of the second visual depiction is performed when the second timepoint is reached. Verification that the second timepoint has been reached is optionally performed by monitoring the audio signal for a second audio tag.

[0291] Optionally, if a lock-on indicator is received, method 1600 further comprises receiving a monitoring indicator and resuming monitoring of the audio signal. The monitoring indicator is indicative of a time difference between reaching the second timepoint and the second visual

depiction or the second subject text being displayed exceeding a threshold. For example, the monitoring indicator is received when a user of the user platform leaving an area of transmission, the area of transmission comprising a geographical location associated with transmission, emission, broadcast, or streaming of the audio signal or the media content.

5 Alternatively, the monitoring indicator is received when the audio signal or the media content is paused, accelerated, decelerated, or otherwise interrupted or interfered with compared to a chronological continuation of the audio signal or the media content.

[0292] Further optionally, method 1600 comprises receiving a user interaction with the user platform associated with the displayed data. For example, the user interaction is indicative of a request for further information associated with the displayed subject data, or indicative of a request to share, review, or otherwise interact with components of the subject file.

10

[0293] Figure 17 illustrates a flowchart of a method for audio track identification and content information transmission at a server device. Specifically, method 1700 is one method performed by components of audio recognition system 400 of Figure 4.

15 [0294] Method 1700 comprises: step 1702, obtaining one or more audio features; step 1704, obtaining a first portion of an audio track; step 1706, extracting one or more audio features from the first portion of the audio track; step 1708, generating a first audio fingerprint comprising the one or more audio features; step 1710, comparing the first audio fingerprint to a plurality of audio fingerprints; step 1712, identifying a best match most similar to the first audio fingerprint; step 1714, transmitting identification data based on the best match; and step 20 1716, transmitting content information associated with the identification data. Steps 1704 and 1006 are alternatives to step 1702, and step 1716 is optional e.g. depending on whether a request for content information is received at the server performing method 1700.

[0295] Step 1702 comprises obtaining, from a user device, one or more audio features extracted from a first portion of an audio track. For example, the server receives one or more audio features extracted by the user device, where the user device comprises a microphone that monitored or otherwise collected an audio signal of the audio track. After receiving one or more audio features, method 1700 proceeds to step 1708. In another example, the server receives a first audio fingerprint generated by the user device, and method 1700 proceeds to step 1710. If the server does not obtain one or more audio features, method 1700 proceeds to step 1704.

25

30

[0296] Step 1704 comprises obtaining a first portion of an audio track, e.g. an audio signal over a time duration associated with an audio track, from the user device.

[0297] Step 1706 comprises extracting the one or more audio features from the first portion. For example, if the user device was unable to extract the one or more audio features from the first portion, the server receives the first portion at step 1704 and extracts the one or more audio features at step 1706. For example, extracting one or more audio features comprises analysing an audio signal of the audio track to identify or quantify selected audio characteristics.

35

[0298] Step 1708 comprises generating a first audio fingerprint comprising the one or more audio features. The first audio fingerprint is associated with information about the audio signal over the first portion, such that the first audio fingerprint is a hashed or otherwise encoded

40

version of the one or more audio features. Not only does the first audio fingerprint require less storage space, but it is also faster to transmit between devices. Moreover, hashing or otherwise encoding the one or more audio features allows for noise, environmental conditions, differences in volume, or other impactors on the audio signal to be filtered out or otherwise compensated for. The first portion of the audio soundtrack played at different volumes from different quality speakers subject to different noise and recorded by different microphones can still result in the same first audio fingerprint.

[0299] Step 1710 comprises comparing the first audio fingerprint to a plurality of audio fingerprints. Each audio fingerprint of the plurality of audio fingerprints is associated with identification data comprising a content identifier and a timepoint, such that each audio fingerprint is associated with a known audio track at a known timepoint. For example, the server compares the first audio fingerprint to the plurality of audio fingerprints stored at a database.

[0300] Step 1712 comprises identifying a best match, where the best match is the most similar audio fingerprint to the first audio fingerprint. For example, a ball-tree algorithm is used at steps 1710 and step 1712 to identify which audio fingerprint of the plurality of audio fingerprints is most similar to, or identical to, the first audio fingerprint. Optionally, step 1710 is repeated for each of the audio fingerprints in the plurality of audio fingerprints until the best match is identified at step 1712. Alternatively, the audio fingerprints are stored or retrieved from a database using a hierarchical structure such that each repetition of step 1710 results in a more similar audio fingerprint being identified. Alternatively, step 1710 and step 1712 are combined, and an artificial intelligence or computational algorithm is applied to extract the most similar audio fingerprint from the plurality of audio fingerprints to the first audio fingerprint, hence identifying the best match. For example, comparing the first audio fingerprint to the plurality of audio fingerprints comprises performing a nearest neighbour search to identify the best match.

[0301] Step 1714 comprises transmitting identification data to the user device based on the best match. For example, the identification data associated with the best match identified at step 1712 is transmitted to the user device, which is used to identify the audio track of steps 1702 – 1006.

[0302] Step 1716 is optional depending on whether a request is received from the user device. If a request is received, step 1716 comprises transmitting content information associated with the identification to the user device. For example, a request for content information is received at the server from the user device, e.g. step 1610 or step 1618 of method 1600, comprising identification data. In response to the request, the server retrieves content information from storage, e.g. a database, associated with the received identification data, e.g. content information relevant to the audio track at the identified timepoint. If a request is not received, step 1716 optionally comprises transmitting a query to the user device to determine whether the audio track was identified successfully based on the best match. Alternatively, step 1716 is not performed.

[0303] Optionally, method 1700 further comprises obtaining a second portion of the audio track, or one or more audio features extracted from the second portion, from the user device. At least steps 1708 – 1714 are then repeated for the second portion. A second audio fingerprint is generated comprising one or more audio features extracted from the second portion. The second audio fingerprint is then compared to the plurality of audio fingerprints to identify an

updated best match, which is the most similar audio fingerprint to the second audio fingerprint. Identification data associated with the updated best match and/or content information associated with the updated best match is then transmitted to the user device. For example, playing of the media content associated with the audio track is paused, skipped, or otherwise interrupted such that it is no longer playing in a typical chronological manner. The second audio fingerprint is then used to maintain synchronisation with the audio track. Alternatively, the second portion of the audio track is a portion of a second audio track, wherein the second audio track is associated with a different media content, e.g. the different media content is now playing on the media playing device. The second audio fingerprint is then used to identify the second audio track.

[0304] As described above, the plurality of audio fingerprints is optionally stored in a database. Preferably, the database is configured such that the database, or a server connected to the database, can retrieve the identification data based on an audio fingerprint, and retrieve the content information based on the identification data.

[0305] Figures 11A and 11B illustrate a flowchart of a method for media content identification and interaction. Specifically, method 1800 is another method performed by components of media interaction system 100, media processing system 200, content synchronisation system 300, or audio recognition system 400 of Figures 1-4. Specifically, method 1800 is an example embodiment of method 1200, although the skilled person would understand that alternative embodiments of method 1200 are possible based on at least the information provided herein.

[0306] Method 1800 comprises: step 1802, obtaining a known media content; step 1804, obtaining a known audio track; step 1806, generating a timeline from the known media content; step 1808, generating an embedded audio track; step 1810, generating content information; step 1812, storing the embedded audio track and the content information; step 1814, monitoring a portion of an unknown audio track; step 1816, identifying the media content by acquiring identification data; step 1818, determining if an audio tag is detected; step 1820, extracting identification data from the audio tag; step 1822, receive identification data from a server; step 1824, initiating a request for content information; step 1826, receiving the request for content information; step 1828, retrieving stored content information using the identification data; and step 1830, outputting the content information to the user device.

[0307] Step 1802 comprises obtaining the known media content associated with a known audio track. If the known audio track is not obtained at step 1802, or does not meet a predetermined quality threshold to allow for optimal audio feature extraction or embedding of audio tags, method 1800 proceeds to step 1804. If the known audio track is obtained at step 1802, method 1800 proceeds to step 1806.

[0308] Step 1804 is optional and comprises obtaining a known audio track. For example, if an audio track obtained at step 1802 was below the predetermined quality threshold, step 1804 comprises retrieving the known audio track and determining that the known audio track exceeds the predetermined quality threshold. E.g., step 1802 and step 1804 are step 1402 of method 1400.

[0309] Step 1806 comprises generating a timeline from the known media content. The timeline comprises a plurality of timepoints, such that the timeline is a sequence of segmented time

intervals for the duration of the known media content. For example, step 1806 is step 1404 of method 1400.

5 [0310] Step 1808 comprises generating an embedded audio track by embedding an audio tag within the known audio track. The audio tag comprises identification data, which is associated with the known media content at a selected timepoint, such as a first timepoint. The audio tag is embedded at frequencies detectable by the microphone and less detectable to a user of the user device. For example, step 1808 is step 1410 of method 1400. Optionally, the embedded audio track is generated by embedding a plurality of audio tags within the known audio track. For example, audio tags are embedded using FSK modulation at regular intervals, e.g. every 10 12 seconds.

[0311] Optionally, step 1808 further comprises generating an audio fingerprint for each timepoint of the timeline, thereby generating a plurality of audio fingerprints. The audio fingerprint comprises one or more audio features extracted from the known audio track at each timepoint. Identification data is linked to each audio fingerprint of the plurality of audio fingerprints based on the plurality of timepoints of the timeline. 15

[0312] Step 1810 comprises generating content information associated with the identification data. Information associated with one or more subject extracted from the known media content at the selected timepoint, e.g. the first timepoint, is collated to form content information. For example, step 1810 is step 1412 and step 1414 of method 1400.

20 [0313] Step 1812 comprises storing the embedded audio track and the content information associated with each timepoint of the timeline. For example, step 1812 is step 1416 of method 1400. Optionally, steps 1802 – 1812 are performed at a workstation or a server.

[0314] Step 1814 comprises monitoring a portion of an unknown audio track using a microphone of the user device. The unknown audio track is associated with the media content emitting from a speaker of a media playing device, which is preferably independent to the user device. Optionally, monitoring the portion comprises monitoring an audio signal, where the audio signal comprises or is otherwise associated with the portion of an unknown audio track, and processing the audio signal to improve detection of the audio tag, e.g. filtering out noise. 25

[0315] In one example, the audio signal further comprises a portion of another unknown audio track. Monitoring the audio signal at step 1814 thus comprises detecting a plurality of audio tags. Step 1820 subsequently comprises extracting identification data for each of the plurality of audio tags, which is associated with a plurality of known media content, and receiving a user interaction selecting the known media content from the plurality of known media content. 30

[0316] Step 1816 comprises identifying the media content by acquiring identification data. If an audio tag is detected at step 1818, step 1816 comprises step 1820. If an audio tag is not detected at step 1818, step 1816 comprises step 1822. For example, steps 1814 – 1822 are steps 1502 – 1508 of method 1500. For example, if an audio tag is detected within the portion of the unknown audio track, identification data is extracted from the audio tag. The extracted identification data is associated with the known media content at a known timepoint, and therefore the unknown audio track is identified as the known audio track of the known media content. 35 40

[0317] Step 1818 comprises determining if an audio tag is detected, e.g. step 1504 of method 1500. If environmental noise is impacting or otherwise affecting the detection of an audio tag required to extract identification data at step 1820, or is impacting or otherwise affecting the extraction of one or more audio features required to receive identification data at step 1822, step 1818 further comprises implementing one or more noise reduction algorithms or signal enhancement techniques. The algorithms/techniques minimise the impact of ambient noise, ensuring reliable audio code detection and feature extraction in noise environments.

[0318] Alternatively, if an audio tag is not detected, step 1822 comprises obtaining one or more audio features extracted from the portion of the unknown audio track monitored by the user device, where the audio features are extracted at the user device or a server. An unknown audio fingerprint comprising the one or more audio features is generated and compared to a plurality of audio fingerprints generated at step 1808. Once the best match is identified, the audio fingerprint most similar to the unknown audio fingerprint, the identification data associated with the best match is transmitted to the user device. Optionally, the server or workstation device is communicatively coupled to a database, such as a database configured to store the embedded audio track, the optional plurality of audio fingerprints, and the content information associated with each timepoint of the timeline.

[0319] Step 1824 comprises initiating, based on a user interaction with the user device, a request for content information from the server device. For example, step 1824 comprises step 1512 of method 1500.

[0320] Step 1826 comprises receiving the request of step 1824 for content information, wherein the request comprises the identification data of step 1816. For example, step 1826 is step 1610 or step 1618 of method 1600.

[0321] Step 1828 comprises retrieving stored content information associated with the identification data. For example, step 1828 is step 1716 of method 1700.

[0322] Step 1830 comprises outputting the content information to the user device, e.g. step 1512 of method 1500.

[0323] Optionally, steps of method 1800 are repeated, e.g. steps 1802 – 1812 are repeated when processing a second known media content or second known audio track. For example, step 1808 repeated comprises embedding, at the workstation device, a second audio tag within a second known audio track at the workstation device. The second audio tag comprises identification data associated with the second known audio track, thereby generating a second embedded audio track. Step 1812 repeated comprises storing the second embedded audio track. Optionally, additional steps are repeated for additional known media content or known audio tracks. For example, step 1814 repeated comprises monitoring, at the user device, a portion of a second unknown audio track using the microphone. Step 1820 repeated comprises extracting the identification data from the second audio tag detected within the portion of the second unknown audio track, thereby identifying the second known audio track as the second known audio track. Step 1824 repeated comprises requesting content information associated with the second known audio track by transmitting the identification data to the server device, and step 1830 repeated comprises receiving, at the user device, content information associated with the second known audio track. Therefore, it is clear to the skilled person that

not all steps of method 1800 are required for the overall process outlined in method 1200 and 1800 to be repeated.

5 [0324] Figure 19 illustrates a flowchart of a method for establishing secure communication at a user device 502. Specifically, Figure 19 illustrates method 1900 performed at a user device 502 and/or user application (referred to here as user device 502) for secure communication and output synchronisation with a transmission device 504, transmission application, and/or SDK as described above (referred to here as transmission device 504).

10 [0325] Method 1900 comprises: step 1902, scan for content; step 1904, detect a first sound code 520; step 1906, convert the first sound code 520 into an identification code 522; step 1908, transmit the identification code 522; step 1910, establishing a bidirectional connection. Method 1900 optionally comprises step 1912, synchronise output.

15 [0326] Step 1902 comprises scanning for content by broadcasting a first request 512 using a bidirectional communication protocol. For example, the user device 502 broadcasts a first request 512, e.g. using WebSockets, to be detected at compatible transmission device 504s 504. A compatible transmission device 504 comprises functionality to receive the first request 512 using the bidirectional communication protocol, such as a device with enabled bidirectional communication protocol access directly or via a transmission application and/or SDK. This allows for bidirectional communication to be encrypted or otherwise configured for secure communication between a user device 502 and a transmission device 504. Optionally, 20 the user device 502 is a user application running on a user device 502 to provide the bidirectional communication functionality, or another functionality described in relation to method 1900.

25 [0327] Step 1904 comprises detecting a first sound code 520 using a microphone. The microphone is controllable from a user application running on a user device 502. Optionally, the microphone is communicatively coupled or otherwise associated with the user device 502. The first sound code 520 comprises an identifier 518 and, optionally, a timestamp, where the identifier 518 is alphanumeric code encoded into sound, e.g. using FSK modulation. The first sound code 520 is emitted from a speaker 508 associated with the transmission device 504, e.g. a speaker controllable from the transmission application/SDK.

30 [0328] Preferably, the first sound code 520 is detectable by the microphone but less detectable to a user of the user device 502. For example, the first sound code 520 uses ultrasound of near-ultrasound frequencies. This allows for sound-based communication between a transmission device 504 and a user device 502 without negatively impacting user interaction with audio output from the transmission device 504, speaker 508, or user device 502.

35 [0329] Step 1906 comprises converting the first sound code 520 into an identification code 522. Conversion comprises decoding the first sound code 520 into alphanumeric code, e.g. reversing a previous encoding process, and/or extracting the identification code 522 from the first sound code 520. Optionally, conversion further comprises decrypting the first sound code 520 and/or alphanumeric code.

40 [0330] Step 1908 comprises transmitting the identification code 522 for authentication using the bidirectional communication protocol. For example, the identification code 522 is transmitted to the transmission device 504 for authentication. Authentication optionally

comprises comparing the identification code 522 with the identifier 518, to determine whether the identification code 522 matches the identifier 518. If the identification code 522 matches the identifier 518, communication with the user device 502 is secure and/or not impacted by error. Otherwise, communication with the user device 502 is either insecure or otherwise impacted by error.

[0331] Step 1910 comprises establishing a bidirectional connection, e.g. a handshake is performed via WebSockets and the bidirectional connection is a full-duplex channel. If the transmission device 504 authenticates the identification code 522, the transmission device 504 and user device 502 are paired using the bidirectional connection, allowing for secure and fast live communication between devices. This allows for efficient and accurate synchronisation of outputs.

[0332] Step 1911 is an optional step comprising synchronising a first output of the user device 502, e.g. output 606 with a second output of the transmission device 504, e.g. output 608. For example, the bidirectional connection is used to communicate timing information between the user device 502 and the transmission device 504.

[0333] In one example, timing information comprises a current timestamp associated with an output at the user device 502, which is identified by the user device 502. A message timestamp is communicated from the transmission device 504 associated with the output at the transmission device 504. If the current timestamp is ahead of than the message timestamp, output at the user device 502 is altered to slow down or otherwise match the timing associated with the message timestamp. Alternatively, if the current timestamp is behind the message timestamp, the output is sped up or otherwise matched. If the current timestamp matches the message timestamp, no action is needed, as the outputs are synchronised.

[0334] In another example, timing information comprises a current timestamp associated with an output at the transmission device 504, which is identified by transmission device 504. A message timestamp is communicated from the user device 502 associated with the output at the user device 502. If the current timestamp is ahead of than the message timestamp, output at the transmission device 504 is altered to slow down or otherwise match the timing associated with the message timestamp. Alternatively, if the current timestamp is behind the message timestamp, the output is sped up or otherwise matched.

[0335] Optionally, one or more instructions to initiate alternation of an output is also communicated. For example, the user device 502 transmits an instruction for the output of the transmission device 504 to speed up based on timing information received from the transmission device 504. In an alternate example, the transmission device 504 transmits an instruction for the transmission device 504 to speed up based on timing information received from the user device 502.

[0336] Preferably, method 1900 is directed to pairing and synchronising a user device 502, e.g. a mobile device, with a transmission device 504 using a bidirectional communication protocol, e.g. WebSockets, and sound, e.g. ultrasound. For example, method 1900 comprises: initiating a scan for content using WebSockets; receiving an ultrasound or near-ultrasound code via a microphone on the mobile device; converting the received ultrasound or near-ultrasound code to a text code; sending the text code via WebSocket to the transmission device 504; performing

a handshake with the transmission device 504 if the text code matches; synchronising the mobile device with the transmission device 504 via WebSockets; and periodically receiving ultrasound or near-ultrasound signals to ensure the mobile device remains in the vicinity of the transmission device 504, and prompting the user to confirm presence if a signal is missed.

5 [0337] Figure 20 illustrates a flowchart of a method for establishing secure communication at a transmission application. Specifically, Figure 20 illustrates method 2000 performed at a transmission device 504, transmission application, and/or SDK as described above (referred to here as transmission device 504) for secure communication and output synchronisation with a user device 502 and/or user application (referred to here as user device 502).

10 [0338] Method 2000 comprises: step 2002, receive a first request; step 2004, transmit a pairing request; step 2006, receive an identifier 518; step 2008, convert the identifier 518 into a first sound code 520; step 2010, receive an identification code 522; step 2012, authenticate; step 2014, perform a handshake; and optional step 2016, error detection.

15 [0339] Step 2002 comprises receiving a first request associated with the user device 502 using a bidirectional communication protocol, e.g. WebSockets. For example, the transmission device 504 receives or otherwise detects the first request broadcast by user device 502 at step 1902 of method 1900.

20 [0340] Step 2004 comprises transmitting a pairing request to a server 516, e.g. sending pairing request 514 to server 516. The pairing request comprises data associated with the first request, such as identification information associated with the user device 502 and/or transmission device 504. The server 516 uses the pairing request to identify authorised devices, trusted devices, and/or historic devices that have paired with transmission device 504 previously.

25 [0341] Step 2006 comprises receiving an identifier 518 from the server 516. For example, the identifier 518 is only received from the server 516 if the server 516 has identified the pairing request, such as by identifying or otherwise authorising the user device 502, a user associated with the user device 502, a location of the user device 502, or other relevant information comprised within/associated with the pairing request. Optionally, identifying the pairing request comprises identifying or otherwise authorising the transmission device 504, a transmission application running or an SDK, e.g. SDK 700.

30 [0342] Step 2008 comprises converting the identifier 518 into a first sound code 520. The first sound code 520 comprises an identifier 518 and, optionally, a message timestamp, where the identifier 518 is alphanumeric code encoded into sound, e.g. using FSK modulation. Optionally, the first sound is emitted from a speaker 508 associated with the transmission device 504, e.g. a speaker controllable from the transmission application/SDK.

35 [0343] Step 2010 comprises receiving an identification code 522 from the user device 502 using the bidirectional communication protocol. For example, the identification code 522 is identification code 522 transmitted at step 1908 of method 1900.

40 [0344] Step 2012 comprises authenticating the identification code 522 based on matching the identifier 518 and the identification code 522. For example, the identification code 522 is compared to identifier 518 to determine whether the user device 502 is authenticated and/or whether the communication with the user device 502 is secure and unimpacted by errors. If

the identification code 522 matches the identifier 518, the user device 502 and communication is secure, authorised/trusted, and/or not impacted by error, and method 2000 proceeds to step 2014. Otherwise, the user device 502 or communication is either insecure, unauthorised/not trusted, or otherwise impacted by error, and method 2000 proceeds to step 2016.

5 [0345] Step 2014 comprises performing a handshake, thereby establishing a bidirectional connection, e.g. the bidirectional connection 702 and/or the bidirectional connection established in step 1910 of method 1900. For example, establishing a bidirectional connection comprises performing a handshake using WebSockets to establish full-duplex connection. Once the WebSocket connection is established, both the paired devices, the user device 502
10 and the transmission device 504, can start exchanging data frames using the WebSocket protocol. The WebSocket protocol provides low-overhead communication, allowing for real-time, bidirectional data exchange, ideal for applications live updates and improved synchronicity between complex and/or data-rich outputs.

[0346] Optionally, the user device 502 transmits a switching request to the transmission device
15 504 to upgrade the bidirectional communication protocol to a bidirectional connection, e.g. the switching request is associated with the identification code 522, to initiate the handshake. The transmission device 504 then responds to the switching request in step 2014, agreeing to upgrade, and a validation response is sent from the user device 502 to the transmission device 504 to confirm the handshake is complete. Further optionally, the first request 512 comprises
20 the switching request and the identification code 522 comprises the validation response.

[0347] Alternatively, step 2014 comprises the transmission device 504 transmitting a switching request to the user device 502 upon authenticating the identification code 522 in step 2012. The user device 502 then responds to the switching request, and the bidirectional connection is established. Further optionally, a validation response is sent from the transmission device
25 504 to the user device 502 to confirm the bidirectional connection.

[0348] Additionally, or alternatively, the server 516 is involved in establishing a bidirectional connection, e.g. the switching request is sent to the server 516 as pairing request 514, which processes the request to determine if the user device 502 and transmission device 504 are authorised/trusted. Optionally, the server 516 determines if both the user device 502 and
30 transmission device 504 have the required functionality to perform the handshake and maintain a bidirectional connection.

[0349] The skilled person would understand there are multiple variations for performing a handshake not limited to the examples detailed above.

[0350] Step 2016 comprises performing error detection, e.g. if the identification code 522 is
35 not authenticated or if the handshake fails.

[0351] For example, if a WebSocket handshake fails, the following error detection and rectification processes can be employed: Retry Mechanism, e.g. implement a retry strategy, where the client attempts the handshake multiple times with a delay between attempts, optionally including an exponential backoff to avoid overwhelming the server 516; Protocol-
40 Level Error Handling, e.g. check for specific status codes or error messages that indicate specific issues that can be logged or trigger alternative actions; TLS/SSL Certificate Validation, e.g. if the handshake fails due to SSL/TLS issues, validate the relevant certificate, where

rectification might involve updating the certificate, adjusting security settings, or notifying the user of the problem; Alternative Connection Methods, e.g. if the handshake consistently fails, attempt alternative protocols or fallback methods (e.g., using HTTP/2 or long-polling) to maintain communication; Network Diagnostics, e.g. perform network diagnostics to check for issues like DNS resolution problems, firewall restrictions, or proxy settings that might block WebSocket connections; User Notification, e.g. notify a user of the handshake failure with specific error details, offering troubleshooting steps or suggesting alternative actions, such as switching networks or contacting support. These methods help address and rectify issues when a WebSocket handshake fails, ensuring potential connectivity problems are managed effectively.

[0352] Preferably, method 2000 is directed to pairing and synchronising a transmission device 504 with a user device 502, e.g. a mobile device, using a bidirectional communication protocol, e.g. WebSockets, and sound, e.g. ultrasound. For example, method 2000 comprises: receiving a pairing request from the mobile device via WebSocket; sending the pairing request to a server 516 to identify the request using transmission ID, user ID, and geo-location; receiving an identifier 518 code from the server 516; converting the identifier 518 code from text to an ultrasound or near-ultrasound code; sending the ultrasound or near-ultrasound code to the mobile device; receiving a text code from the mobile device via WebSocket; performing a handshake with the mobile device if the text code matches; and periodically sending ultrasound or near-ultrasound signals to ensure the mobile device remains in the vicinity of the transmission device 504.

[0353] In one optimised example, methods 1900 and 2000 are combined, e.g. using system 500. Optionally, to enable pairing and synchronisation using WebSockets and ultrasound: a transmission device comprises a secured/authorised WebSocket access via either an SDK or a transmission application as described above; the transmission device is configured for sending sound or information, data, or instructions to another associated device to send sound upon request; and/or the user device runs a user application capable of picking up sound via a microphone.

[0354] For example, according to one embodiment, the user application initiates scanning for content using WebSockets. The transmission device with WebSocket access (TDwWS) receives this request and sends a pairing request to a service-embedded SDK. The SDK forwards the request to a server, which identifies the request using transmission ID, user ID, and/or geo-location. The server then sends an identifier code to the SDK. This code is transformed from text to an ultrasound or near-ultrasound code by the SDK, which then sends the ultrasound or near-ultrasound code. The user application picks up this code via the device's microphone and converts it back to text. The user application sends the text code via WebSocket to the TDwWS. If the text code matches, the WebSockets perform a handshake, and the devices are now paired. Synchronisation is then performed via WebSockets. To ensure the consumer remains in the vicinity of the TDwWS, ultrasound or near-ultrasound signals are sent at intervals. If a signal is missed, the application prompts the user with a message, e.g. "Are you still watching?" The user can confirm by interacting with a confirmation selection in the user application.

[0355] Further implementations optionally includes error correction or redundancy in the ultrasound signal to improve reliability in noisy environments. An optional robust encoding scheme like Reed-Solomon helps ensure the ultrasound code is accurately received and decoded. To maintain session management and security, ultrasound or near-ultrasound codes are optionally sent at intervals to either revoke or prolong access to content synchronisation. A grace period or retry mechanism is optionally implemented before revoking access if a signal is missed. Additionally, geo-tagging ensures the connection is broken when the user leaves the vicinity of the TDwWS.

[0356] The SDK for the hosting service handles receiving pairing requests, transforming the identifier code into ultrasound, and maintaining the WebSocket connection. The SDK is configured for receiving pairing requests, transforming the identifier code to ultrasound, sending the ultrasound signal, WebSocket communication, and timestamping. Functionality provided by the SDK includes starting a WebSocket connection with timestamp synchronisation, handling pairing requests, sending ultrasound signals with timestamps, sending requests to the server, retrieving and/or identifying a current timestamp, and handling WebSocket messages with timestamps.

[0357] A user application is configured to receive the ultrasound signal and WebSocket messages, extract timestamps, and synchronise accordingly. The user application is configured for receiving pairing requests, transforming ultrasound to identifier code, sending the identifier code, extracting timestamps, synchronising accordingly, and WebSocket communication. Functionality provided by the user application includes starting a WebSocket connection and handling timestamp synchronisation, sending scan requests with a current timestamp, initiating listening for ultrasound signals, processing recorded ultrasound data and extracting timestamps, synchronising based on timestamps, sending decoded codes via WebSocket, handling WebSocket messages with timestamps, and/or retrieving a file associated with recognised audio and the current timestamp.

[0358] For synchronisation, both the SDK and user application embed timestamps in WebSocket messages and ultrasound signals. This allows both devices to synchronise based on the timing of the content being played. The user application adjusts its synchronisation based on the difference between the received timestamp and its current timestamp, ensuring that content-related actions on the user device align with output associated with the SDK. Periodic heartbeat messages are exchanged to continuously synchronise the two devices, including timestamps to correct any drift over time.

[0359] Security and reliability are enhanced by ensuring both the SDK and user application use the same time source, such as a network time protocol (NTP), to maintain consistency in timestamps. Error correction is optionally implemented for ultrasound signals, and the WebSocket connection optionally secure, e.g. using WSS. Mechanisms for connection recovery are also preferably implemented to handle disruptions.

[0360] Figures 21A and 21B illustrate a system architecture diagram for a video interaction system. Specifically, Figure 21A illustrates system 2100 for processing media content, e.g. video content comprising an audio track and accompanying visual elements, and Figure 21B illustrates system 2100 for synchronising with media content.

[0361] System 2100 comprises a media content 2102, a time interval 2104, a media segment 2106, a workstation device 2108, a hierarchical hash structure 2110, a database 2112, a transmission device 2114, a user device 2116, and data 2118.

5 [0362] A media content 2102, e.g. video content, is divided based on time intervals 2104 into media segments 2106, each a predefined duration such as 1 second long. Optionally, the predefined duration is selected based on balancing time efficient content synchronisation with accurate identification of media segments. The media segment 2106 is hashed, preferably using cryptographic hashing techniques or algorithms, e.g. SHA-256. A hierarchical hash structure 2110 such as a Merkle tree is constructed using the hashes, where each leaf node
10 represents a segment hash and parent nodes are hashes of combined, e.g. concatenated, child hashes, culminating in a root hash. The root hash is distributed to the user device 2116, preferably when synchronisation with the media content 2102 begins, and is optionally stored using blockchain technology. The database 2112 comprises data, e.g. metadata, associated with the media content 2102. For example, descriptions, subjects, or other embedded elements
15 described above. The metadata is prestored in the database, such that content associated with a specific media segment 2106 is retrievable based on the segment hash.

[0363] The user device 2116, e.g. a user application installed or otherwise operating from a mobile, tablet, or electronic device, which is optionally the same device as the workstation device 2108, also hashes a media segment 2106, which is obtained by monitoring the media
20 content 2102 being played, transmitted, or is otherwise available from the transmission device 2114. Optionally, the media content 2102 is playing or otherwise being displayed on the user device 2116, e.g. transmission device 2114 is user device 2116. The segment hash is used by the user device 2116 to retrieve data 2118 associated with the media segment 2106, e.g. content information for the media content 2102 and the media segment 2106 specifically, such
25 as the location(s) shown in media content 2102 during the time interval 2104.

[0364] The system 2100 is configured to handle various scenarios where parts of the media content 2102 may be missing, corrupted, or unrecognized. In such cases, the user device 2116 employs algorithms such as Dynamic Time Warping (DTW) and/or AI-based prediction models to adjust timestamps and predict hashes for the missing segments. This ensures continuous
30 and seamless synchronization of the media content 2102, even in the presence of disruptions.

[0365] One of the beneficial advantages of system 2100 is its ability to provide secure and efficient synchronization between devices and media content. The use of cryptographic hashing and hierarchical hash structures ensures that the media content 2102 is tamper-proof and its integrity is maintained throughout the transmission and playback process. Additionally,
35 the integration of metadata from the database 2112 allows for a rich and interactive user experience, as the user device 2116 can dynamically retrieve and display relevant information based on the media segment 2106 being played.

[0366] Cryptographic hashing, as applied to media segments 2106, involves the use of secure hashing algorithms to generate a unique fixed-size hash value for each segment of the media
40 content 2102. The process begins by dividing the media content 2102 into discrete, fixed-duration segments, such as 1-second intervals. Each segment is then processed through a cryptographic hash function, such as SHA-256, which produces a 256-bit hash value. This hash

value serves as a unique identifier for the media segment 2106, ensuring that even the slightest alteration in the segment's content will result in a significantly different hash value.

5 [0367] The use of cryptographic hashing provides several beneficial advantages. Firstly, it ensures data integrity, as any modification to the media segment 2106 will be immediately detectable through a mismatch in the hash values. Secondly, it enhances security by making it computationally infeasible to reverse-engineer the original media content 2102 from the hash value, thereby protecting the content from unauthorized access or tampering. Additionally, cryptographic hashing facilitates efficient data comparison and retrieval, as hash values can be quickly compared to verify the authenticity and integrity of the media segments.

10 [0368] Alternative cryptographic hashing algorithms, such as SHA-3 or Blake2, can also be employed depending on specific security requirements and computational efficiency considerations. These algorithms offer varying levels of security and performance, allowing for flexibility in implementation based on the needs of the system 2100. For example, SHA-3, being part of the NIST standard, provides a higher level of security assurance, while Blake2 is
15 optimized for faster performance and lower computational overhead.

[0369] The hierarchical hash structure 2110, preferably implemented as a Merkle tree, is constructed using the cryptographic hashes of the media segments 2106. In this structure, each leaf node represents the hash of an individual media segment 2106. Parent nodes are generated by concatenating the hashes of their child nodes and then hashing the concatenated
20 result. This process continues up the tree until a single root hash is obtained, representing the entire set of media segments 2106.

[0370] The Merkle tree structure offers several significant advantages. It enables efficient and secure verification of the integrity of the entire media content 2102. By distributing the root hash to the user device 2116 at the start of content consumption, the system ensures that the
25 user device can verify the authenticity of any media segment 2106 by traversing the Merkle tree and comparing the computed segment hash with the corresponding leaf node. This hierarchical approach also allows for partial verification, where only a subset of the media segments needs to be verified, significantly reducing the computational load.

[0371] Moreover, the hierarchical hash structure 2110 supports efficient data synchronization and resynchronization. In scenarios where parts of the media content 2102 are missing, corrupted, or unrecognized, the system can quickly identify and isolate the affected segments
30 by comparing their hashes with the Merkle tree. This facilitates targeted resynchronization efforts, such as re-transmitting only the affected segments or using AI-based prediction models to estimate the missing hashes.

35 [0372] Alternative hierarchical structures, such as hash lists or hash chains, can also be considered. While these alternatives may offer simpler implementations, they typically lack the scalability and efficiency of Merkle trees in handling large datasets and supporting partial verification. However, in scenarios with less stringent security requirements or smaller datasets, these alternatives may provide a viable solution.

40 [0373] The combination of cryptographic hashing and hierarchical hash structures in system 2100 ensures robust, secure, and efficient synchronization of media content 2102. These techniques provide a comprehensive framework for verifying data integrity, facilitating dynamic

metadata integration, and enabling seamless user experiences across various content consumption scenarios.

5 [0374] AI predictive hashes are an optional technique employed to maintain synchronization and data integrity in scenarios where parts of the media content 2102 are missing, corrupted, or unrecognized. This method leverages artificial intelligence and machine learning algorithms to predict the hash values of these problematic segments based on the context and content of adjacent recognized segments. The primary objective of AI predictive hashes is to ensure continuous and seamless synchronization, even in the presence of data anomalies or transmission errors.

10 [0375] The process of generating AI predictive hashes begins with the identification of unrecognized or missing media segments 2106. Once these segments are detected, the system analyses the surrounding recognized segments to gather contextual information. Machine learning models, such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs), are then employed to predict the hash values of the unrecognized segments.
15 These models are trained on extensive datasets comprising various media content, enabling them to accurately infer the likely hash values based on patterns and correlations observed in the data.

[0376] The use of AI predictive hashes offers several beneficial advantages. Firstly, it enhances the robustness of the synchronization process by providing a mechanism to handle data anomalies without interrupting the user experience. Secondly, it improves the accuracy of synchronization by leveraging the predictive capabilities of AI to generate hash values that closely match the original content. Additionally, AI predictive hashes reduce the need for retransmission of missing segments, thereby optimizing bandwidth usage and minimizing latency.

25 [0377] Alternative approaches to AI predictive hashes include heuristic-based methods and statistical models. Heuristic-based methods rely on predefined rules and patterns to estimate the hash values of unrecognized segments. While these methods are simpler to implement, they may lack the accuracy and adaptability of AI-based approaches. Statistical models, such as Markov chains or Bayesian networks, can also be used to predict hash values based on probabilistic relationships between segments. However, these models may require extensive computational resources and may not perform as well in complex or highly variable content scenarios.

30 [0378] By leveraging the predictive power of machine learning algorithms, this technique ensures seamless and accurate synchronization, even in the presence of data anomalies. The flexibility and adaptability of AI predictive hashes make them a valuable addition to the overall synchronization framework, complementing other methods such as cryptographic hashing and hierarchical hash structures.

40 [0379] In one example, the process of implementing AI predictive hashes begins with the collection of a comprehensive and diverse dataset. This dataset encompasses a wide range of audio and video content, including movies, TV shows, music videos, commercials, and standalone audio tracks. The goal is to capture various contexts and scenarios in which media content is consumed, suitably mirroring the eventual use-case breadth where possible. The

dataset should be annotated with metadata, including segment hashes, timestamps, and contextual tags (e.g., whether a song is part of a movie or a standalone track). The data should be collected in a format that is compatible with machine learning frameworks, such as CSV files for metadata and binary files for media content. The size of the dataset is large enough to ensure that the AI models can learn the intricate patterns and correlations within the media content. Typically, a dataset comprising several terabytes of annotated media content is recommended to achieve high accuracy in predictions.

[0380] Selecting an appropriate machine learning model is beneficial for the success of AI predictive hashes. Given the sequential nature of media content, recurrent neural networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), are well-suited for this task. These models excel at capturing temporal dependencies and patterns within sequential data. Additionally, convolutional neural networks (CNNs) can be employed to extract spatial features from video frames, which can be combined with the temporal features extracted by RNNs. A hybrid model that integrates both RNNs and CNNs can provide a comprehensive understanding of the media content, enabling accurate hash predictions. The model architecture should be designed to handle the high dimensionality of video data and the sequential nature of audio data.

[0381] Training the AI model involves feeding the collected dataset into the selected model architecture. The training process is iterative and requires significant computational resources, typically involving the use of GPUs or TPUs to accelerate the training. The model is trained to minimize the loss function, which measures the difference between the predicted hashes and the actual hashes of the media segments. Techniques such as backpropagation and gradient descent are used to update the model parameters. The training process also involves data augmentation techniques, such as adding noise or altering the playback speed, to improve the model's robustness and generalization capabilities. The training phase may take several days to weeks, depending on the size of the dataset and the complexity of the model.

[0382] Once the model is trained, it is beneficial to verify its accuracy and reliability. This involves evaluating the model on a separate validation dataset that was not used during training. The validation dataset should be representative of the real-world scenarios in which the model will be applied. Metrics such as mean squared error (MSE), mean absolute error (MAE), and accuracy are used to assess the model's performance. Additionally, the model's predictions are compared against the ground truth hashes to ensure that the predicted hashes are within an acceptable range of the actual hashes. Cross-validation techniques, such as k-fold cross-validation, can be employed to further validate the model's performance and ensure that it is not overfitting to the training data.

[0383] After successful verification, the AI model is deployed within the system 2100 to generate predictive hashes in real-time. During content playback, the user device 2116 monitors the media content 2102 and computes segment hashes. If a segment is unrecognized or missing, the AI model is invoked to predict the hash based on the context provided by adjacent recognized segments. The predicted hash is then used to retrieve the corresponding metadata from the database 2112, ensuring continuous synchronization and a seamless user experience. The system dynamically adjusts the synchronization using the predicted hashes, leveraging techniques such as dynamic time warping to align the timestamps accurately. The

AI model continuously learns and improves over time by incorporating feedback from real-world usage, further enhancing its predictive capabilities.

5 [0384] The implementation of AI predictive hashes comprises a comprehensive process of data collection, model selection, training, verification, and application. By leveraging advanced machine learning techniques and extensive datasets, the system ensures robust and accurate synchronization of media content, even in the presence of data anomalies. This approach enhances the overall user experience by providing seamless and secure media content interaction.

10 [0385] Beneficially, system 2100 is configured to distribute the root hash of the hierarchical hash structure 2110 to the user device 2116. The distribution of the root hash ensures the integrity and synchronization of media content within the proposed system. The root hash, derived from the hierarchical hash structure (e.g. Merkle tree), serves as a unique and secure identifier for the entire set of media segments. The primary purpose of distributing the root hash is to provide a reference point for verifying the authenticity and integrity of the media content during playback. This process begins at the start of content consumption, where the root hash is transmitted to the user device 2116. The distribution can be achieved through various secure channels, such as encrypted communication protocols, the bidirectional communication described above e.g. WebSockets, or secure token-based systems. By providing the root hash at the outset, the system 2100 ensures that the user device 2116 can continuously verify the integrity of each media segment 2106 by comparing the computed segment hashes with the corresponding nodes in the Merkle tree. This mechanism not only enhances security but also facilitates efficient synchronization, as any discrepancies or tampering with the media content 2102 can be immediately detected and addressed.

25 [0386] Dynamic metadata matching is an optional process that enhances the interactivity and contextual relevance of media content during playback. The system 2100 leverages a pre-made database 2112 that stores metadata 2118 associated with various media segments 2106, such as descriptions, products, locations, and other associated elements. During playback, the user device 2116 computes the hash of each media segment 2106 and uses this hash to query the database 2112 for the corresponding metadata 2118. This dynamic retrieval ensures that the metadata 2118 is precisely synchronized with the specific video segments 2106 being played. The system 2100 optionally employs algorithms to match the computed segment hashes with the pre-stored metadata, enabling real-time updates and interactions. For example, if a particular scene in a video features a product, the system can dynamically display relevant information or links to purchase the product.

35 [0387] Context differentiation is a beneficial aspect of the described systems throughout the description, ensuring that the metadata and synchronization processes accurately reflect the specific usage scenarios of the media content. Context differentiation mechanisms described in relation to system 2100 are therefore applicable to all systems described herein. The metadata database 2112 includes contextual tags that differentiate between various contexts in which the content is used. For example, a song may be used as part of a movie, a commercial, or as standalone audio on a radio or streaming service. By tagging the content with these contextual markers, the system can provide precise differentiation and tailored interactions based on the specific context. This differentiation is achieved through a

40

combination of metadata tagging and hierarchical hash relationships. The system can identify whether a song is embedded within a larger piece of content, such as a movie, or if it is being played independently. This ensures that the user receives contextually relevant information and interactions, enhancing the overall experience. Additionally, context differentiation allows for accurate tracking and reporting of content usage across different platforms, providing valuable insights for content creators and advertisers.

[0388] Hierarchical context association is a method used to maintain the relationships between different levels of content within the proposed system. This approach involves linking the hashes of standalone media segments, such as songs, with the hashes of the larger content in which they are embedded, such as movies or commercials. By creating these hierarchical associations, the system can accurately reflect the context in which the content is used. For example, a song that appears in a movie will have its hash linked to the overarching movie hash, allowing the system to differentiate between the song's standalone usage and its embedded usage within the movie. This hierarchical structure ensures that the metadata and synchronization processes are contextually accurate and relevant. The system can dynamically adjust the interactions and information displayed to the user based on the specific context, providing a seamless and engaging experience. Additionally, hierarchical context association facilitates efficient content management and tracking, enabling content creators and distributors to monitor and analyse the usage of their content across different platforms and contexts. This approach enhances the robustness and scalability of the system, ensuring that it can handle a wide range of media types and usage scenarios.

[0389] Optionally, user device 2116 uses a sliding hash window technique to enhance the robustness and efficiency of the synchronization process. This approach involves continuously re-hashing overlapping segments of the media content to facilitate quick resynchronization in the event of data loss or corruption. The sliding window operates by moving incrementally across the media content, hashing each segment and its overlapping neighbours. For example, if the predefined segment duration is 1 second, the sliding window might hash segments from 0 to 1 second, 0.5 to 1.5 seconds, and so on. This overlapping ensures that even if a segment is missed or corrupted, the adjacent overlapping segments can still be used to verify and maintain synchronization. The sliding hash window technique significantly reduces the computational overhead by avoiding the need to re-hash the entire content, focusing only on the affected segments. This method is particularly useful in dynamic streaming environments where network conditions may vary, ensuring that the user experience remains seamless and uninterrupted.

[0390] Recognition algorithms are an optionally integrated to system 2100, enabling real-time identification of objects, faces, and other subjects within video content 2102. These algorithms leverage advanced machine learning and computer vision techniques to analyze and interpret the visual and auditory elements of the media. The primary goal of recognition algorithms is to enhance user interaction and synchronization by dynamically associating recognized subjects with relevant metadata. For example, facial recognition algorithms can identify actors in a movie, while object recognition algorithms can detect products or locations. These algorithms typically involve training deep learning models, such as CNNs and RNNs, on large annotated datasets. Examples include the use of YOLO (You Only Look Once) for real-time object detection and OpenFace for facial recognition. The implementation of these algorithms

requires substantial computational resources and careful tuning to achieve high accuracy and low latency. By integrating recognition algorithms, the system can provide enriched metadata, enabling features such as interactive product placements, contextual advertisements, and enhanced content recommendations.

5 [0391] Metadata enrichment is an optional process of enhancing basic metadata 2118 associated with media content 2102 by incorporating additional contextual and dynamic information. This process leverages the capabilities of recognition algorithms and pre-made databases to provide a richer and more interactive user experience. For example, when a recognition algorithm identifies a specific product in a video, the system 2100 can automatically
10 retrieve and display detailed information about the product, such as its name, price, and purchase links. Similarly, facial recognition can be used to provide background information about actors or characters in a scene. Metadata enrichment also involves dynamically updating the metadata based on user interactions and real-time content analysis. This ensures that the metadata remains relevant and engaging, adapting to the specific context and preferences of
15 the user. The enriched metadata can be stored in a structured format, such as JSON or XML, to facilitate efficient retrieval and integration with the media content. By providing enriched metadata, the system enhances the overall user experience, making the content more informative, interactive, and engaging.

[0392] End-to-end encryption is a beneficial security measure optionally implemented in the
20 system, or any other system. to ensure the confidentiality and integrity of hash lists and metadata. This encryption technique involves encrypting data at the source and decrypting it only at the intended destination, preventing unauthorized access or tampering during transmission. The system employs asymmetric encryption algorithms, such as RSA (Rivest-Shamir-Adleman), to securely encrypt the hash lists and metadata. In this process, a public
25 beneficial is used to encrypt the data, while a corresponding private beneficial is used for decryption. This ensures that only authorized devices with the correct private beneficial can access and decrypt the data. End-to-end encryption provides robust protection against various security threats, including man-in-the-middle attacks and data breaches. Additionally, it ensures that the integrity of the hash lists and metadata is maintained, as any tampering with
30 the encrypted data would result in decryption failures. By implementing end-to-end encryption, the system guarantees that the synchronization process remains secure and trustworthy, safeguarding the privacy and integrity of the media content and associated metadata.

[0393] System 2100 offers significant advantages for streaming platforms by ensuring real-time synchronization of interactive features with video playback. The system's ability to
35 segment video content into fixed-duration parts and hash each segment allows for precise synchronization of metadata and interactive elements. For example, during a live sports event, the system can dynamically display player statistics, game highlights, and advertisements in sync with the video stream. The dual synchronization mechanism, which combines ultrasound metadata and cryptographic hashing, ensures that synchronization is maintained even in the
40 presence of network fluctuations or data loss. This robust synchronization capability enhances the user experience by providing seamless and interactive content consumption, making streaming platforms more engaging and informative.

[0394] System 2100 can revolutionize e-commerce integration by seamlessly linking product placements in video content with actionable consumer interactions. By leveraging the dynamic metadata matching feature, the system can identify products featured in video segments and retrieve relevant information from a pre-made database. For example, during a fashion show live stream, the system can display product details, prices, and purchase links for the clothing items worn by models. The hierarchical context association ensures that products are accurately tagged whether they appear as standalone items or as part of a larger context, such as a movie or commercial. This capability not only enhances the viewer's experience but also provides a direct and efficient pathway for e-commerce transactions, driving sales and increasing revenue for content creators and advertisers.

[0395] In the gaming industry, system 2100 ensures synchronized in-game actions with live-streamed video content, providing a more immersive and interactive experience for players. The system's ability to hash video segments and dynamically match metadata allows for real-time updates and interactions based on the game's progress. For example, during a live-streamed eSports tournament, the system can display player profiles, game statistics, and sponsor advertisements in sync with the gameplay. The sliding hash window technique facilitates quick resynchronization in case of data loss or network issues, ensuring that the synchronization remains intact throughout the live stream. This capability enhances the viewer's engagement and provides valuable insights and information, making the gaming experience more enjoyable and interactive.

[0396] System 2100 can significantly enhance educational content by providing interactive video tutorials with synchronized question-answer segments. The system's ability to segment and hash video content ensures that educational videos are precisely synchronized with supplementary materials, such as quizzes, annotations, and interactive exercises. For example, during an online lecture, the system can dynamically display relevant questions and prompts based on the video content being played. The context differentiation feature ensures that the educational content is accurately tagged and differentiated based on its usage scenario, whether it is part of a larger course or a standalone tutorial. This capability enhances the learning experience by providing interactive and contextually relevant content, making education more engaging and effective.

[0397] System 2100 optionally comprises live subject recognition, which can enhance interactivity with real-time identification and interaction with subjects in video content. By implementing AI-based recognition algorithms, the system 2100 can identify objects, faces, and other subjects within the video and dynamically associate them with enriched metadata. For example, during a live concert stream, the system can identify the performing artists and display their biographies, discographies, and social media links in real-time. This capability not only enhances the viewer's experience but also provides valuable information and interactions, making live content more engaging and informative. The system's ability to handle unrecognized parts using AI prediction and dynamic time-warping algorithms ensures that synchronization remains seamless, even in the presence of data anomalies.

[0398] System 2100 ensures accurate context differentiation and seamless recognition for scenarios where a song transitions between embedded usage and standalone play. The system uses temporal hash clustering to detect whether the song hash appears as part of a

broader context or as an independent playback. For example, if a song hash aligns with movie content hashes, it is tagged as embedded. If it appears without parent context hashes, it is tagged as standalone. This capability ensures that the system accurately differentiates between the various usage scenarios, providing precise synchronization and metadata matching. The dual synchronization mechanism, which includes ultrasound metadata as a fallback system, ensures continuous synchronization even in cases where hashing might be temporarily disrupted. This robust and flexible synchronization capability enhances the user experience by providing seamless and contextually accurate content consumption.

[0399] Figure 22 illustrates a system architecture diagram of a dual synchronisation system. Specifically, figure 22 illustrates dual synchronisation system 2200, comprising components of system 2100 in addition to speaker 2202 and audio tag 2204. The speaker 2202 is configured to emit audio signals associated with media content 2102, e.g. speaker 2202 is communicative connected or integral to transmission device 2214, and the audio tag 2204 is embedded in the audio signal as described in detail above. Beneficially, the dual synchronisation system is configured to dynamically switch between video hashing of system 2100 and ultrasound synchronisation of e.g. content synchronisation system 300.

[0400] Switching within the dual synchronization system 2200 is a beneficial mechanism that ensures continuous and seamless synchronization of media content, even in the presence of disruptions or varying network conditions. The system dynamically switches between video hashing, as implemented in system 2100, and ultrasound synchronization, as utilized in content synchronization system 300. This switching process is initiated based on predefined criteria, such as the integrity of the hash verification process or the presence of ultrasound metadata. When the video hashing mechanism detects discrepancies or fails to verify the integrity of the media segments due to missing or corrupted data, the system seamlessly transitions to ultrasound synchronization. This is achieved by leveraging the audio tag 2204 embedded in the audio signal emitted by the speaker 2202. The audio tag preferably provides a coarse-grain synchronization reference, allowing the system to maintain alignment with the media content. The primary benefit of this dynamic switching capability is the fail-safe mechanism it provides, ensuring that synchronization is maintained even under adverse conditions. This enhances the robustness and reliability of the system, delivering an uninterrupted user experience.

[0401] Error detection is a fundamental aspect of the dual synchronization system 2200, ensuring the integrity and accuracy of the synchronization process. The system optionally employs multiple layers of error detection mechanisms to identify and address discrepancies in the media content. The primary method involves the use of cryptographic hashing, where each media segment is hashed, and the resulting hash values are compared against the hierarchical hash structure (e.g. Merkle tree). Any mismatch between the computed hash and the expected hash indicates a potential error, prompting the system to initiate corrective measures. Additionally, the system incorporates error correction algorithms, e.g. Reed-Solomon, to detect and correct minor data losses during video transmission. These algorithms add redundancy to the data, allowing the system to recover lost or corrupted segments. The dual synchronization mechanism further enhances error detection by providing an alternative synchronization reference through ultrasound metadata. If the video hashing process fails to verify the integrity of the media segments, the system switches to ultrasound synchronization, ensuring that errors are detected and addressed promptly. This multi-layered approach to

error detection enhances the reliability and accuracy of the synchronization process, ensuring a seamless user experience.

5 [0402] The system 2200 optionally employs one or more advanced techniques to prevent and detect tampering. As well as the use of cryptographic hashing, where each media segment is hashed using secure algorithms such as SHA-256 among other methods, the system stores the root hashes using blockchain technology or distributed ledgers. This ensures that the root hashes are immutable and tamper-proof, providing a trustworthy reference for verification. Additionally, the system employs end-to-end encryption to protect the hash lists and metadata during transmission, preventing unauthorized access or tampering. By combining these
10 advanced security measures, the dual synchronization system 2200 ensures that the media content remains secure and trustworthy, providing a robust and reliable synchronization framework.

[0403] Scalability is a beneficial design consideration for systems 2100 and 2200, ensuring that the system can efficiently manage large volumes of data and support a growing number
15 of users. The system architecture is configured to handle various media types and usage scenarios, leveraging distributed computing resources to process and store the hash lists and metadata. The hierarchical hash structure (e.g. Merkle tree) and sliding hash window technique minimize computational overhead, allowing the system to scale efficiently. Additionally, the use of cloud-based storage and processing solutions enables the system to dynamically allocate
20 resources based on demand, ensuring that it can handle peak loads and large-scale deployments. This scalability ensures that the system can provide robust and reliable synchronization across diverse content platforms and user bases.

[0404] Figure 23 illustrates a flowchart of method for processing video content. Specifically, the flowchart illustrates method 2300 for processing known media content.

25 [0405] The method 2300 begins with step 2302, which involves segmenting a known media content into a plurality of media segments. This is achieved by dividing the media content into fixed-duration segments, such as 1-second intervals. The predetermined duration is selected to balance the need for efficient synchronization with the requirement for accurate identification of media segments. This segmentation process ensures that each segment is manageable in
30 size and can be individually processed, hashed, and verified. The segmentation is performed at a workstation device, which may be a server or a dedicated processing unit, ensuring that the media content is systematically divided into discrete, time-bound segments.

[0406] In step 2304, each media segment of the plurality of media segments is hashed using
35 cryptographic algorithms, such as SHA-256. This hashing process generates a unique hash value for each segment, referred to as a segment hash. The segment hash serves as a digital fingerprint for the media segment, ensuring that even the slightest alteration in the segment's content will result in a significantly different hash value. Each segment hash is associated with the known media content at a specific time interval, providing a precise and secure method for verifying the integrity and authenticity of the media content. The use of cryptographic hashing
40 ensures that the segment hashes are tamper-proof and can be reliably used for synchronization and verification purposes.

[0407] Step 2306 comprises arranging the plurality of segment hashes on one or more lower hierarchical levels of a hierarchical hash structure, such as a Merkle tree. In this structure, each leaf node represents a segment hash. The hierarchical hash structure is configured by combining hash values from the lower hierarchical levels to generate hash values of higher hierarchical levels. This process involves concatenating the hash values of child nodes and hashing the concatenated result to produce the parent node's hash value. This hierarchical arrangement continues until a single root hash is generated, representing the entire set of media segments. The root hash serves as a unique identifier for the known media content over all time intervals, providing a secure and efficient method for verifying the integrity of the entire media content.

[0408] In step 2308, the plurality of segment hashes is associated with data comprising one or more context tags. These context tags are used to differentiate between known media contents and provide additional metadata for each segment. The context tags may include information such as the type of content (e.g., movie, commercial, standalone audio), the specific usage scenario (e.g., part of a larger content or standalone), and other relevant metadata (e.g., descriptions, products, locations). This association ensures that each segment hash is contextually enriched, allowing for precise differentiation and dynamic metadata matching during playback. The context tags are stored in a pre-made database, enabling the system to retrieve and display relevant information in real-time, enhancing the user experience and providing valuable insights into the media content.

[0409] Method 2300 ensures robust, secure, and efficient processing of media content using hashing at a workstation device. The segmentation, hashing, hierarchical arrangement, and contextual association of media segments provide a comprehensive framework for verifying the integrity, authenticity, and contextual relevance of media content, ensuring seamless synchronization and enhanced user interaction. Optionally, method 2300 comprises additional steps from any previous method, specifically methods relating to processing of media content for integration or synchronisation.

[0410] Figure 24 illustrates a flowchart of a method for synchronising based on video hashing. Specifically, a flowchart of method 2400 for synchronising display on a user interface of a user device based on video hashes generated by a user application installed on the user device. Receiving a root hash

[0411] The method 2400 begins with step 2402, which involves receiving a root hash of a hierarchical hash structure associated with a media content. This root hash is a unique identifier that represents the entire set of media segments, ensuring the integrity and authenticity of the media content. The hierarchical hash structure, such as a Merkle tree, is constructed by combining hash values from lower hierarchical levels to generate hash values of higher hierarchical levels, culminating in the root hash. The root hash is transmitted to the user device at the start of content consumption, providing a reference point for verifying the integrity of the media content throughout playback.

[0412] At step 2404, the user device retrieves data from a database associated with the root hash. This data comprises a contextual tag for differentiating between known media contents and content information for media content interaction. The contextual tag provides information about the specific usage scenario of the media content, such as whether it is part of a movie,

a commercial, or standalone audio. The content information includes metadata such as descriptions, products, locations, and other embedded elements that enhance the user experience. By associating the root hash with this data, the system ensures that the media content is contextually enriched and ready for dynamic interaction during playback.

5 [0413] Step 2406 comprises monitoring the media content at a first timestamp for a predetermined duration, thereby obtaining a first media segment. The predetermined duration is typically a fixed interval, such as 1 second, ensuring that each segment is manageable in size and can be individually processed. The user device continuously monitors the media content, capturing segments at specific timestamps to facilitate synchronization and
10 verification. This step ensures that the media content is systematically divided into discrete, time-bound segments that can be hashed and compared against the hierarchical hash structure.

[0414] In step 2408, the user device verifies synchronization by computing a hash of the first media segment and comparing the hash with the hierarchical hash structure. The computed
15 hash is matched against the corresponding hash in the Merkle tree to verify the integrity and authenticity of the media segment. If the computed hash matches the expected hash, synchronization is verified, indicating that the media content has not been tampered with or corrupted. This verification process ensures that the media content remains secure and trustworthy throughout playback.

20 [0415] If synchronization is verified, the method proceeds to step 2410, where the user device uses the hash to retrieve and display content information associated with the first timestamp. This content information is dynamically matched and retrieved from the pre-made database, providing users with relevant metadata and interactive elements. For example, if the media segment corresponds to a scene in a movie, the system can display information about the
25 actors, locations, or products featured in that scene. This dynamic retrieval and display of content information enhance the user experience by providing contextually relevant and interactive content.

[0416] If synchronization is not verified, the method proceeds to performing corrective measures to maintain synchronization. In step 2412, the user device adjusts the first timestamp
30 for resynchronizing and displaying content information associated with the adjusted first timestamp. This adjustment involves using algorithms such as DTW to align the timestamps accurately, ensuring that the media content remains synchronized despite any discrepancies. By adjusting the timestamp, the system can recover from minor synchronization errors and continue providing a seamless user experience.

35 [0417] Alternatively, or additionally, to step 2412, in step 2414 the system can use a trained machine learning model to generate a predicted hash based on one or more adjacent media segments. This approach leverages AI-based prediction to estimate the hash of the unrecognized or corrupted segment, enabling the system to retrieve and display content information associated with the predicted timestamp. The machine learning model is trained
40 on simulated or historical datasets, allowing it to accurately predict hashes based on the context provided by adjacent segments. This predictive capability ensures continuity and synchronization even in the presence of data anomalies.

[0418] Step 2416 comprises monitoring the media content at a second timestamp for the predetermined duration, with the second timestamp within the predetermined duration from the first timestamp. This step ensures that the system captures overlapping media segments, facilitating continuous monitoring and synchronization. By obtaining a second media segment
5 that overlaps with the first media segment, the system can maintain synchronization and verify the integrity of the media content more effectively.

[0419] At step 2418, the user device computes a hash of the second media segment, thereby re-hashing overlapping media segments for maintaining synchronization. This sliding hash window approach ensures that the system continuously verifies the integrity of the media
10 content, even if certain segments are missed or corrupted. By re-hashing overlapping segments, the system can quickly resynchronize and maintain a seamless user experience. This method ensures robust and reliable synchronization, providing users with secure and interactive media content consumption.

[0420] Systems 2100 and 2200, along with methods 2300 and 2400, are preferably connected,
15 forming a comprehensive framework for media content synchronization. System 2100 focuses on the hashing of media content, dividing it into fixed-duration segments, hashing each segment, and constructing a hierarchical hash structure (e.g. Merkle tree) to ensure secure and efficient synchronization. Method 2300 details the steps involved in this process, from segmenting the media content to associating segment hashes with contextual metadata.
20 System 2200 builds upon system 2100 by incorporating a dual synchronization mechanism that dynamically switches between video hashing and ultrasound-based synchronization. This system includes additional components such as a speaker and audio tag, which emit and decode ultrasound signals for synchronization. Method 2400 outlines the steps for maintaining
25 synchronization at a user device, including receiving the root hash, monitoring media content, verifying synchronization, and handling unrecognized segments through AI prediction and timestamp adjustment. Together, these systems and methods provide a robust, secure, and interactive framework for media content synchronization, ensuring seamless user experiences across various platforms and scenarios.

[0421] Systems 2100 and 2200, along with methods 2300 and 2400, are closely related to
30 systems and methods for ultrasound signal synchronization, e.g. systems and methods comprising audio tag based processes. The dual synchronization mechanism in system 2200 leverages ultrasound metadata as a fallback system to maintain coarse-grain synchronization when video hashing encounters issues. This integration ensures continuous synchronization by dynamically switching between ultrasound-based and hash-based synchronization. The
35 ultrasound signals, emitted by the speaker and decoded by the user device, provide a reliable synchronization reference, complementing the cryptographic hashing approach. This combination enhances the robustness and reliability of the synchronization process, ensuring that media content remains synchronized even in the presence of data anomalies or network fluctuations.

[0422] The disclosed systems and methods also relate to audio hashing and audio fingerprint-based synchronization. The hashing techniques used in systems 2100 and 2200, and detailed
40 in methods 2300 and 2400, can be applied to audio content in addition to video content, or any other form of media content, such as paintings in a gallery, where synchronisation is based on

position rather than time. By segmenting audio content, hashing each segment, and constructing a hierarchical hash structure, the system can achieve precise synchronization of audio content across various platforms. Similar to audio fingerprinting, where unique identifiers are generated for audio segments to facilitate synchronization and identification, all described process can be combined for a comprehensive content integration and synchronisation system. For example, communication between devices is performed using bidirectional communication protocols as described for secure retrieval of content information 2118 based on hashes, audio fingerprints, or audio tags. The integration of AI-based prediction and dynamic metadata matching further enhances the system's capability to handle unrecognized audio segments and provide contextually relevant information. This comprehensive framework ensures accurate and reliable synchronization of both audio and video content, leveraging the strengths of cryptographic hashing and audio fingerprinting techniques.

[0423] Optionally, the disclosed systems herein comprise one or more AI models, where the one or more AI models are integrated in a structured pipeline to enhance the extraction, analysis, and synthesis of content-related information. Unlike conventional systems that operate AI models in isolation, each AI model within this system functions as a modular component in a larger network, allowing extracted features from one model to inform subsequent models. This architecture not only improves accuracy and efficiency but also generates new, derived insights that would be difficult to obtain through independent AI models.

[0424] For example, in the context of media content identification, a primary AI model optionally first extracts audio features from a broadcasted media segment using frequency analysis, spectral decomposition, or waveform analysis techniques. These extracted features, which may include spectral peaks, temporal energy distribution, or Mel-frequency cepstral coefficients (MFCCs), are then fed into a secondary AI model trained for feature matching against a database of known media segments. In scenarios where an exact match is not found, an additional AI model leveraging machine learning-based inference, such as a convolutional neural network (CNN) or transformer-based architecture, may be applied to identify the closest match by considering temporal patterns, phonetic similarities, or semantic correlations.

[0425] This cascading approach allows for improved robustness in identifying media content even in challenging environments, such as those with background noise or partial audio obfuscation. Moreover, the system benefits from self-learning capabilities, where feedback from incorrect matches or user interactions refines the performance of the AI models over time. This can be achieved through reinforcement learning techniques or adaptive retraining mechanisms that prioritize frequently encountered misidentifications for further fine-tuning.

[0426] Beyond media identification, the system optionally comprises content synchronization and information retrieval. A tertiary AI model may be tasked with extracting relevant contextual elements from the identified media, such as objects, people, or locations appearing in a scene. This is particularly relevant for enhancing interactive experiences in applications such as smart television systems, second-screen experiences, or augmented reality (AR) overlays. For example, once a media segment is identified, an AI-driven entity recognition model can analyse subtitles, closed captions, or visual metadata to determine key subjects or thematic elements. If a film scene contains a recognizable landmark, such as the Eiffel Tower, the AI model can

link this visual entity to a structured knowledge graph, retrieving supplementary data about its history, significance, or related cultural references.

[0427] The scalability and modular nature of the system are further reinforced by its loosely coupled architecture, which allows for seamless expansion and adaptability across different domains. Each AI component functions independently while still contributing to the unified workflow, ensuring that individual models can be updated or replaced without disrupting the overarching system. This is particularly advantageous when integrating new AI capabilities, such as sentiment analysis, speech-to-text conversion, or real-time translation, enabling a continuously evolving framework that remains state-of-the-art without requiring complete system overhauls.

[0428] An alternative implementation of this AI integration pipeline is in security and authentication applications, where AI models work in tandem to verify user identity based on biometric or behavioural data. For instance, a first AI model could process voice biometrics from a media stream, while a secondary AI model cross-references this data with stored voice signatures. If discrepancies are detected, an additional AI model employing anomaly detection techniques could analyse contextual factors, such as background noise or user intent, to improve verification accuracy. This approach ensures multi-layered security, reducing false positives and improving overall system resilience.

[0429] Additionally, the system leverages bidirectional communication and data exchange mechanisms to optimize performance. This is particularly evident in real-time content synchronization, where timestamp alignment between media playback and supplementary content must be maintained. In this context, an AI model monitors real-time variations in playback speed, buffering events, or network latency, dynamically adjusting the retrieval and presentation of related content. For example, if a live sports broadcast experiences a delay due to streaming congestion, the AI system can recalibrate secondary content, such as player statistics or event highlights, to maintain contextual relevance.

[0430] Moreover, the integration of hierarchical hash structures and cryptographic techniques enhances the system's reliability in verifying content authenticity and mitigating potential tampering. AI models can generate unique audio-visual fingerprints for each media segment, embedding them within a cryptographic chain to ensure that retrieved content aligns with the originally indexed reference. This has critical applications in media rights management, digital watermarking, and forensic content tracing.

[0431] From a deployment perspective, the system is designed to operate efficiently on distributed architectures, enabling cloud-based processing as well as edge computing scenarios. This flexibility allows for computationally intensive AI tasks, such as deep learning inference or large-scale data matching, to be offloaded to cloud servers, while lightweight AI models handling real-time interaction can be executed directly on user devices. This hybrid deployment strategy optimizes both latency and computational efficiency, ensuring that high-priority tasks are handled with minimal delay while resource-intensive operations benefit from cloud scalability.

[0432] In an example optional use case comprising interactive media content discovery, a user may utilize a smartphone application to identify a television show playing in the background.

The system captures an audio snippet, extracts key features, and queries a remote database to retrieve potential matches. If an exact match is found, the application presents the user with contextual information, such as cast details, trivia, and product placements. If no direct match is available, the AI system suggests alternative results ranked by probability, providing the user with informed options rather than an outright failure. Furthermore, user interactions—such as confirming or rejecting a suggested match—are logged to refine the AI model's accuracy over time.

[0433] Another alternative implementation extends to smart home environments, where AI models orchestrate cross-device synchronization between multimedia devices. For example, if a user begins watching content on a television and subsequently moves to another room, the AI-driven system can facilitate seamless playback continuation on a secondary device, adjusting audio and visual parameters to account for differing speaker configurations and display settings, e.g. using any of the methods, systems or processes described above.

[0434] Beneficially, the disclosed system represents a significant advancement in AI-driven content interaction by seamlessly integrating multiple AI tools in a scalable, modular, and loosely coupled framework. This integration enhances content identification, synchronization, and contextual information retrieval, while ensuring adaptability across diverse applications. By enabling AI models to dynamically interact and refine their outputs based on multi-modal inputs, the system achieves higher accuracy, improved user engagement, and enhanced operational efficiency. These optional applicability across a broad range of industries, from entertainment and media to security, e-commerce, and smart home automation.

[0435] The present disclosure extends the previously disclosed media interaction system by incorporating hybrid radio metadata enrichment, bidirectional device synchronization, and AI-driven subject extraction for radio broadcasts. The combination of real-time and pre-programmed metadata retrieval methods ensures accurate, dynamic content engagement while maintaining security, efficiency, and low-latency processing. The disclosed system further leverages previously described bidirectional synchronization techniques to optimize metadata alignment and listener interaction across live and archived radio content. These advancements support a unified, scalable approach to interactive radio content consumption, reinforcing the principles of seamless media interaction, secure communication, and synchronized metadata enrichment.

[0436] For example, a bidirectional connectivity between the workstation, AI components, and RadioDNS is established to enhance both live and pre-programmed radio broadcasts. This integration relies on several key protocols and technologies. RadioDNS Hybrid Radio provides metadata linking for live radio, while the Service & Programme Information (SPI - ETSI TS 102 818) enables broadcasters to structure metadata about stations, shows, and content. The RadioEPG (Electronic Program Guide - ETSI TS 102 818) synchronizes pre-programmed content with metadata, and RadioTAG (ETSI TS 103 270) allows listeners to tag audio content for future interaction, whether live or on-demand. Additional technologies include DAB Slideshow & RadioVIS (ETSI TS 101 499), which support visual enhancements like ads, images, and contextual metadata, as well as RTSP and HLS protocols for streaming radio, ensuring synchronized metadata overlays.

[0437] The workstation and AI preferably interacts with both live and pre-programmed radio in multiple ways. Metadata enrichment for live radio utilizes RadioDNS SPI and AI-enhanced tagging to pull in real-time metadata, whereas pre-programmed radio relies on RadioEPG to pre-load metadata. Content recognition for live radio involves AI listening to audio fingerprints and hashing real-time streams, while for pre-recorded content, AI scans files for tagging and enrichment. Interactive features such as look-up, purchasing, booking, sharing, commenting, betting, and rating are available in real-time via an SDK linked to RadioTAG, and for pre-programmed content, scheduled and stored interactions are supported, similar to podcast engagement. Optionally the system comprises integration of live commerce and/or real-time advertisement insertion using RadioVIS and RadioEPG, while pre-tagged content facilitates e-commerce and booking integrations. Additionally, a dashboard for content creators and media owners provides a live interface for real-time enrichment and content synchronization, with AI-driven recommendations for pre-programmed tagging.

[0438] To effectively manage both live and pre-programmed radio, the workstation dashboard preferably comprises the following features. Real-time content enrichment for live radio involves an AI listener that monitors live DAB+ or streaming radio feeds and matches audio fingerprints with metadata. RadioDNS API integration fetches station metadata, program details, and service information, while auto-syncing tags and contextual data enhance content with AI-generated topics, keywords, and commerce links. RadioTAG event handling enables users to tag moments in broadcasts for later engagement, and automated call-to-action (CTA) triggers introduce interactive options, such as "Buy Now," "Book," or "Learn More." Additionally, real-time subject insertion allows enhanced content control.

[0439] For pre-programmed content management, the dashboard integrates RadioEPG to load scheduled program metadata. Batch processing of pre-recorded content uses AI to scan the database for relevant subjects, and a marketplace linkage pre-assigns subjects to pre-recorded material. A hybrid interaction management system ensures seamless switching between live and pre-programmed content, maintaining a consistent interactive experience regardless of format. An API for third-party app integration enables connections with radio applications, automotive dashboards, and smart home devices. Furthermore, user behavior analytics and reporting features track listener engagement, conversions, and the success of various interactions.

[0440] The technical workflow is preferably divided into two primary processes: live radio and pre-programmed radio. In the live radio flow, the RadioDNS API fetches station and program metadata, while AI listens and matches content by identifying relevant topics, people, products, and locations. The workstation then synchronizes commerce, ratings, and user interactions, allowing listeners to buy, book, comment, share, and bet in real-time through a consumer-facing app. In the pre-programmed radio flow, RadioEPG pre-loads scheduled content, and AI scans audio segments to tag contextually relevant information. Interactive elements are pre-loaded to enable engagement after broadcast, allowing users to re-interact with archived content, purchase products, book experiences, and participate in other post-broadcast activities.

[0441] Further aspects are described in detail below.

[0442] An aspect of the present disclosure provides a modular and scalable approach to integrating metadata enrichment, real-time content interaction, and secure device synchronization for both live and pre-programmed radio broadcasts. The integration leverages hybrid radio standards and streaming protocols, facilitating enriched listener engagement through bidirectional communication between a workstation platform, an artificial intelligence-based enrichment module, and an interactive media retrieval system.

[0443] According to an aspect of the present disclosure, there is provided a system for integrating live and pre-programmed radio content with real-time metadata extraction, content enrichment, and interactive engagement. The system is configured to establish a bidirectional connection between a workstation device and a hybrid radio metadata provider. The system utilizes a secure communication protocol to ensure synchronized retrieval and transmission of station metadata, program information, and enriched content subject data between the workstation device and user devices. This facilitates real-time synchronization, interaction, and commerce-enablement for radio broadcasts, regardless of whether the content is consumed live or accessed as an archived recording.

[0444] For example, a workstation device configured for hybrid radio integration optionally receives live metadata streams from a hybrid radio metadata provider via the RadioDNS API. This metadata includes station identification data, program schedules, content descriptions, and real-time audio tagging information. The workstation device processes this data in conjunction with an AI-driven content enrichment module, which dynamically links the metadata with contextual subjects, such as artists, topics, products, locations, or events mentioned within the broadcast. This enriched metadata is then synchronized with the live radio stream and transmitted to user devices via a bidirectional communication protocol.

[0445] In one optional implementation, a DAB+ digital radio receiver within the workstation device continuously retrieves metadata from an ETSI TS 102 818-compliant Electronic Programme Guide (RadioEPG), providing a structured timeline of upcoming programs. This timeline is preloaded into the workstation, allowing the AI enrichment module to pre-tag subjects associated with scheduled content before the program airs. This pre-tagging process enhances interactive engagement, enabling users to access additional contextual information, such as biographies, product links, or historical references, without requiring post-broadcast processing.

[0446] As an alternative, in environments where DAB+ signals are unavailable or unreliable, the system optionally utilize HLS or RTSP streaming protocols to ingest radio streams from internet-based sources. In this case, audio fingerprinting technology is applied to the incoming stream, matching content against a database of pre-indexed radio segments. If a match is detected, the system retrieves corresponding metadata from the workstation database and aligns it with the current broadcast timeline. This fallback mechanism ensures that metadata enrichment remains available even in scenarios where direct integration with a hybrid radio provider is not possible.

[0447] Another example of the system's optional operation is in the context of live listener interaction via the RadioTAG standard (ETSI TS 103 270). When a listener engages with a broadcast by tagging a moment of interest, the system transmits this tag via the bidirectional communication protocol to the central workstation. The workstation then processes the tag

and retrieves associated metadata, allowing the listener to access additional details, save content for later, or trigger interactive actions such as purchasing a product mentioned in the broadcast, booking a related event, or subscribing to updates from the broadcaster. In a collaborative setting, multiple listeners optionally tag the same segment, enabling real-time aggregation of audience interactions and enhancing broadcaster analytics.

[0448] In an alternative implementation, the system optionally extends its functionality beyond real-time broadcasts to include on-demand and archived radio content. For pre-programmed content, the AI enrichment module scans audio files prior to publication, extracting speech-to-text transcriptions, keyword associations, and contextual subject mappings. These enriched segments are then stored within the workstation database, ensuring that users engaging with archived content receive the same depth of interactivity as those listening to live broadcasts.

[0449] Beneficially, by implementing hybrid radio metadata integration with real-time AI-based enrichment, the system significantly enhances listener engagement, content accessibility, and broadcaster commercial opportunities. The seamless synchronization between metadata providers, AI processing modules, and user-facing applications ensures that relevant content is surfaced to users in real time, minimizing delays and maximizing interactivity. By supporting both live and pre-programmed radio, the system provides a consistent and interactive experience, regardless of when the content is consumed.

[0450] The use of bidirectional communication protocols ensures that metadata retrieval, content synchronization, and listener interactions are maintained in real time. Unlike traditional radio metadata systems, which are often one-directional and limited to static program guides, the system disclosed herein allows for continuous two-way engagement between broadcasters and listeners. This is particularly advantageous in applications where commerce integration, audience analytics, and dynamic content adaptation are required.

[0451] Furthermore, the modular nature of the system allows for seamless integration with third-party platforms, automotive infotainment systems, and smart home devices. By supporting standardized hybrid radio protocols (e.g., SPI, RadioEPG, and RadioTAG) alongside AI-driven metadata enrichment, the system ensures that broadcasters and content creators optionally maximize the value of their content across multiple distribution channels.

[0452] Additionally, the system's fallback mechanisms, including audio fingerprinting and indexed metadata retrieval, provide resilience against connectivity issues, regional broadcasting limitations, and metadata inconsistencies. By leveraging a combination of embedded metadata retrieval and audio-based content recognition, the system guarantees high-accuracy synchronization, even in challenging broadcast environments.

[0453] In summary, the integration of hybrid radio metadata and interactive media enrichment provides a robust, scalable, and future-proof solution for enhancing live and on-demand radio experiences. Whether implemented for traditional broadcast environments, digital streaming platforms, or hybrid radio ecosystems, the system ensures that metadata-rich, interactive, and commerce-enabled content engagement is delivered to users in a seamless and intuitive manner.

[0454] Optionally, the system implements a bidirectional communication protocol that enables seamless synchronization between live and pre-programmed radio broadcasts and enriched

metadata for interactive engagement. Hybrid radio metadata is retrieved via a service-oriented interface that integrates with external metadata providers, such as those adhering to ETSI specifications for RadioDNS Hybrid Radio. Specifically, the system supports any one or more of:

5 [0455] Service & Programme Information (SPI - ETSI TS 102 818): Provides structured metadata about radio stations, programs, and content scheduling.

[0456] RadioEPG (Electronic Program Guide - ETSI TS 102 818): Facilitates pre-loading of metadata for pre-programmed content, allowing contextual enrichment and interaction scheduling.

10 [0457] RadioTAG (ETSI TS 103 270): Enables real-time listener tagging of audio segments for future engagement, supporting interactive elements across both live and archived radio.

[0458] DAB Slideshow & RadioVIS (ETSI TS 101 499): Enhances content with synchronized visual metadata, such as images, advertisements, and interactive subject overlays.

15 [0459] RTSP & HLS Streaming Protocols: Supports synchronized audio stream processing, ensuring metadata overlays are consistently aligned with real-time broadcast timing.

[0460] The bidirectional communication protocol further ensures that the system operates independently of proprietary streaming services and optionally retrieve station metadata from open, standardized metadata sources. Secure WebSockets and other communication mechanisms (e.g., TLS/SSL-encrypted WebRTC channels) provide secure and latency-
20 optimized bidirectional communication between the workstation platform, metadata servers, and user devices.

[0461] For example, a workstation platform configured for bidirectional hybrid radio metadata integration optionally establishes real-time communication channels with external metadata providers using a service-oriented interface. This interface facilitates seamless retrieval, processing, and synchronization of station metadata, program schedules, enriched subject
25 information, and interactive content tags. The workstation platform acts as an intermediary processing hub, ensuring that live and pre-programmed radio broadcasts are contextually enriched with metadata while maintaining synchronization across broadcast streams and user interactions.

30 [0462] In an example implementation, Service & Programme Information (SPI - ETSI TS 102 818) is leveraged to structure and retrieve station metadata, enabling the workstation to automatically detect and categorize incoming radio streams based on station identity, program details, and content schedules. When a user interacts with a radio broadcast, the retrieved SPI metadata is linked to a subject-specific knowledge graph, allowing AI-driven enrichment to
35 dynamically associate relevant topics, people, locations, and commerce opportunities. This approach ensures that even dynamically scheduled or regionally varied content maintains interactive metadata consistency.

[0463] Alternatively, for pre-programmed content, the system integrates RadioEPG (Electronic Program Guide - ETSI TS 102 818) to preload metadata for upcoming broadcasts. This enables
40 AI-driven contextual enrichment before a program airs, ensuring that listeners engaging with scheduled content receive pre-tagged metadata, structured annotations, and interactive

elements. The RadioEPG framework also supports metadata caching, allowing the system to operate efficiently in scenarios with intermittent connectivity or where real-time metadata retrieval is constrained.

5 [0464] A key feature of the system is real-time user interaction via RadioTAG (ETSI TS 103 270), which allows listeners to tag specific audio segments for later engagement. When a listener activates an interaction (e.g., bookmarking a music track, identifying a spoken topic, or saving a product mention), the RadioTAG protocol transmits this engagement data via the bidirectional communication layer. The workstation then processes and enriches these tagged moments, retrieving subject metadata, contextual links, and transactional elements (e.g.,
10 purchase options, event bookings, or social engagement prompts). This interactivity is maintained for both live and archived radio broadcasts, ensuring that historical listener engagement data remains actionable.

[0465] In another implementation, visual enhancements for hybrid radio content are enabled through DAB Slideshow & RadioVIS (ETSI TS 101 499). These protocols support synchronized
15 image overlays, advertisements, and program-related visuals that correspond to the broadcast audio. The system ensures alignment of visual metadata with radio content playback, allowing broadcasters to deliver contextually relevant images, logos, sponsor promotions, and dynamically generated UI elements to listeners in real-time. This visual layer is particularly effective in automotive infotainment systems, smart displays, and mobile radio applications,
20 where graphical enhancements optionally increase engagement and provide additional commercial or integration/interaction opportunities.

[0466] To support internet-based streaming workflows, the system integrates RTSP (Real-Time Streaming Protocol) and HLS (HTTP Live Streaming), ensuring low-latency audio stream processing with synchronized metadata overlays. For example, an HLS-enabled radio stream
25 optionally transmits chunked audio segments, with the system processing each segment's associated metadata, visual elements, and interactive tags before it reaches the end-user. This streaming architecture allows for adaptive bitrate streaming, ensuring that metadata enrichment remains functional even in bandwidth-constrained environments.

[0467] Beneficially, implementing a bidirectional communication protocol with hybrid radio
30 metadata support ensures high-accuracy metadata synchronization, real-time listener engagement, and enriched interactivity across both live and pre-programmed radio broadcasts. The system's independence from proprietary streaming services enables seamless integration with multiple hybrid radio providers, broadcasters, and third-party platforms, ensuring that metadata interoperability and content accessibility are maximized.

35 [0468] By utilizing secure WebSockets and encrypted WebRTC channels, the system maintains low-latency, secure data transmission, preventing unauthorized metadata interception, data tampering, or spoofed content manipulation. Unlike conventional metadata retrieval mechanisms that rely on polling-based HTTP requests, the persistent WebSocket connection ensures continuous metadata flow between the workstation platform and listener
40 devices, minimizing latency while maximizing metadata update frequency.

[0469] Additionally, the modular integration of hybrid radio standards ensures that metadata retrieval and synchronization workflows remain adaptable to various broadcasting

infrastructures. Whether implemented for terrestrial DAB+ radio, internet-based streaming, or hybrid radio ecosystems, the system's flexible metadata processing engine ensures that content remains enriched, interactive, and synchronized across diverse radio consumption environments.

5 [0470] Furthermore, the system's ability to process metadata in real time while maintaining bidirectional interactivity introduces new opportunities for broadcasters, advertisers, and content creators. Interactive commerce models, such as real-time product tagging, event-based advertising, and audience analytics tracking, are seamlessly integrated into the broadcast experience, enabling enhanced revenue generation and audience retention
10 strategies.

[0471] In summary, by implementing a service-oriented interface for bidirectional hybrid radio metadata retrieval and enrichment, the system optimizes radio content synchronization, enhances listener engagement, and provides a scalable framework for interactive broadcast applications. Whether deployed for live radio, scheduled content, or archived broadcasts, the
15 system ensures that metadata-rich, contextually enhanced, and commerce-enabled radio engagement is delivered to audiences efficiently and securely.

[0472] According to another optional aspect of the present disclosure, a machine learning-driven content recognition system processes live and pre-programmed radio content to enhance metadata retrieval and listener engagement. The system incorporates an artificial
20 intelligence model configured to perform any one or more of:

[0473] Process Real-Time Audio Streams: The AI model listens to live radio streams via DAB+ or network-based streaming protocols (e.g., HLS, RTSP). It extracts audio fingerprints and matches them with metadata retrieved from the hybrid radio metadata provider.

[0474] Perform Audio Fingerprint Matching: Similar to previously disclosed content
25 synchronization methods, an audio fingerprint is generated for radio segments, enabling real-time content identification and metadata retrieval even in the absence of embedded metadata tags.

[0475] Auto-Sync Tags and Contextual Data: The AI model dynamically associates topics, keywords, and commerce links with the radio content in real time, providing automatic
30 enrichment of broadcast data.

[0476] AI-Powered Subject Extraction: By analysing audio patterns, speech recognition outputs, and embedded metadata, the system extracts relevant subjects from live and archived content. These subjects include products, people, locations, and contextual elements, enhancing the depth of available metadata.

35 [0477] The AI model(s) is further integrated with machine learning algorithms optimized for noisy environments, ensuring high-accuracy identification and synchronization of content metadata, even in cases where metadata transmission is inconsistent or disrupted. This approach mirrors the previously disclosed dual-method strategy for media identification and synchronization, which combines embedded metadata retrieval (where available) with fallback
40 fingerprint-based recognition.

[0478] For example, an AI-driven content recognition system deployed within a workstation platform continuously monitors live and pre-recorded radio streams, leveraging DAB+ broadcast signals or internet-based audio streaming protocols such as HLS (HTTP Live Streaming) and RTSP (Real-Time Streaming Protocol). The system processes these streams in real time, extracting audio fingerprints, speech-to-text transcriptions, and embedded metadata markers. These extracted features are then analysed against a pre-indexed metadata database populated via a hybrid radio metadata provider. By correlating incoming audio with known reference segments, the system is capable of identifying radio programs, categorizing spoken content, and linking contextual subjects to each detected segment.

[0479] In one implementation, the AI model applies advanced digital signal processing (DSP) techniques to transform raw audio streams into structured feature representations. These representations include spectrogram analysis, pitch detection, harmonic structure modelling, and time-frequency decomposition. Using these extracted features, an audio fingerprint is generated that uniquely identifies the segment, allowing for real-time program recognition. Unlike traditional metadata tagging approaches, which rely solely on pre-embedded identifiers, this approach ensures that even untagged or previously unindexed radio content optionally be recognized, classified, and enriched with relevant metadata.

[0480] As an alternative, in scenarios where embedded metadata tags (e.g., RadioTAG markers, DAB+ service labels, or in-stream captions) are present, the system prioritizes direct metadata extraction while simultaneously cross-validating the accuracy of retrieved metadata using audio fingerprinting as a redundancy measure. This hybrid approach improves fault tolerance by ensuring that metadata retrieval remains functional even in environments with inconsistent or missing data transmissions.

[0481] To further optimize listener engagement, the system performs real-time topic detection and contextual keyword analysis, associating relevant commerce links, event references, or supplementary information with the identified audio segments. For example, if the system detects a music track playing on a radio station, it optionally auto-sync metadata such as the artist's biography, album details, or live concert availability. If the detected content relates to a spoken segment (e.g., a sports discussion or interview), the system retrieves and overlays relevant statistics, links to merchandise, or upcoming match details.

[0482] An important feature of the system is AI-powered subject extraction, which leverages machine learning-driven speech recognition and natural language processing (NLP) models to extract named entities, objects, and thematic topics from radio broadcasts. The AI model is trained on multi-domain datasets, enabling it to accurately recognize persons, locations, organizations, products, and trending topics within speech-based content. For instance, if a radio host mentions a film title, celebrity name, or political event, the system automatically associates relevant metadata, linking it to news articles, official profiles, or related commerce integrations.

[0483] Additionally, the AI model is optimized to function in high-noise environments, employing adaptive noise suppression, robust speech isolation, and dynamic feature weighting to maintain high-accuracy identification even when background noise, overlapping speech, or poor transmission quality affects the audio signal. This ensures that content recognition

remains reliable across various broadcast conditions, such as mobile radio playback, in-car listening, or low-bitrate streaming.

5 [0484] Beneficially, by implementing machine learning-driven content recognition, the system achieves high-precision metadata enrichment, seamless content categorization, and enhanced user interactivity. The ability to match real-time broadcasts against structured metadata sources allows for intelligent content discovery and enhanced personalization, ensuring that listeners receive contextually relevant information in real time.

10 [0485] Unlike traditional manual tagging and post-broadcast metadata processing, which require significant human intervention and delayed content indexing, the AI-driven system provides automated, real-time enrichment, making it ideal for dynamic, fast-paced broadcasting environments such as news radio, live sports coverage, and talk radio formats.

15 [0486] Furthermore, the dual-method approach, combining audio fingerprinting and embedded metadata retrieval, ensures data integrity, redundancy, and adaptability to varying broadcast conditions. This approach extends beyond traditional metadata enrichment by enabling persistent listener engagement with both live and archived content, regardless of whether metadata was initially embedded in the stream.

20 [0487] Additionally, the machine learning architecture optionally supports self-improving recognition models, which adapt over time based on listener interactions, broadcaster updates, and new content ingestion. This enables the system to continuously refine its accuracy, expand its content knowledge base, and enhance its contextual awareness, ensuring that future metadata retrieval remains increasingly precise and comprehensive.

25 [0488] By integrating machine learning-based audio analysis, fingerprinting, and metadata synchronization, the system establishes a next-generation content recognition framework that transforms traditional radio broadcasting into an interactive, data-rich, and commerce-enabled experience. Whether implemented for live radio, scheduled programming, or archived content discovery, the system ensures that metadata-enriched, AI-powered radio engagement remains seamless, accurate, and fully interactive across all listening environments.

30 [0489] According to another optional aspect of the present disclosure, there is provided a method for enabling interactive listener engagement for live and pre-programmed radio broadcasts. The method comprises:

[0490] Real-Time Interactions via RadioTAG and Bidirectional Communication – A listener using a user device optionally interact with live radio content via an SDK-enabled interface. RadioTAG functionality allows users to tag audio segments, which are then transmitted to a server device for retrieval and enrichment.

35 [0491] Pre-Programmed Interaction via RadioEPG – For scheduled or archived content, metadata is preloaded into a workstation device, allowing AI-driven tagging and interaction assignment before the content is aired or accessed. The workstation device integrates this metadata with commerce-enabled actions, such as purchasing a referenced product or booking an advertised service.

40 [0492] Marketplace Integration for Commerce-Enabled Content – Whether the interaction occurs live or post-broadcast, pre-tagged content enables streamlined e-commerce, direct

response advertising, and contextual commerce. Interactive subjects (e.g., “Buy Now,” “Book,” “Learn More”) are inserted dynamically based on AI-driven metadata enrichment.

5 [0493] The system ensures that listener interactions are seamlessly synchronized with radio content, regardless of whether the engagement occurs in real time or during a later playback session. This aligns with the previously disclosed methodologies for pairing and synchronization, where bidirectional communication is leveraged to sustain persistent content metadata associations.

10 [0494] For example, a listener engaging with a live radio broadcast via a user device optionally tags specific moments in the stream using an SDK-enabled interface that integrates with RadioTAG (ETSI TS 103 270). This interaction is facilitated by a bidirectional communication protocol, ensuring that tagged segments are immediately transmitted to a central server for processing, enrichment, and future retrieval. If a listener hears an interview with an artist or a discussion about a product, they optionally tag that moment, which triggers an automated metadata lookup. The system retrieves relevant contextual information—such as artist details, product purchase links, or related articles—and presents it within the user interface in real time.

20 [0495] In one implementation, RadioTAG event handling is augmented with AI-driven content segmentation, allowing the system to automatically identify natural breakpoints within broadcasts, such as transitions between news segments, advertisements, or music tracks. This enables users to interact with specific portions of the audio stream more precisely, improving content navigation and engagement accuracy. Additionally, crowdsourced tagging is supported, where multiple listeners tagging the same segment creates a popularity-weighted metadata index, improving searchability and recommended content discovery.

25 [0496] Alternatively, in pre-programmed or archived radio broadcasts, the system integrates RadioEPG (ETSI TS 102 818) to preload metadata and schedule interactive elements before airing. The AI-driven workstation analyses upcoming content, assigns predictive tags, and links e-commerce, booking, or information retrieval actions to specific segments. For instance, if a scheduled program includes a discussion about an upcoming music festival, the system pre-associates ticket purchase links with that segment, ensuring that when the program airs, listeners optionally seamlessly book tickets directly through the interface.

30 [0497] To extend interactivity beyond live engagement, the system supports persistent metadata associations for archived content. Listeners accessing previously aired content via an on-demand platform optionally revisit tagged moments, triggering the same contextual metadata retrieval, purchasing options, or related content recommendations as if they had engaged during the live broadcast. This method effectively blurs the boundary between live and pre-recorded radio experiences, ensuring that interactivity and commerce-driven engagement remain seamless across different listening contexts.

35 [0498] A key aspect of the system is marketplace integration, where commerce-enabled actions are dynamically inserted into radio content based on AI-driven metadata enrichment. Whether interactions occur in real time or are initiated during a later playback session, the system ensures that contextually relevant commerce opportunities—such as product

purchases, ticket bookings, subscription sign-ups, or event reservations—are available at the precise moment they are relevant to the listener.

[0499] For example, when a listener tags a segment discussing a new tech gadget, the system optionally retrieve and display product purchase links from an integrated e-commerce platform. Similarly, if a radio interview promotes an upcoming concert, a “Book Now” button is automatically inserted into the user interface, allowing instant ticket reservations without requiring manual searching. This integration significantly enhances broadcaster-user interaction opportunities, as advertisers and content creators optionally ensure that commercial interactions remain timely and highly contextualized.

[0500] Beneficially, the system’s bidirectional communication protocol ensures that listener interactions are fully synchronized with radio content, eliminating latency issues, mismatched metadata, or content desynchronization. Unlike traditional radio-based advertising and commerce, which often rely on manual user searches or delayed follow-ups, this approach enables immediate and frictionless engagement, enhancing conversion rates and user experience.

[0501] Additionally, the integration of RadioTAG, RadioEPG, and marketplace commerce functionality creates a cohesive and modular framework for interactive radio engagement, applicable to music, talk shows, live news, sports commentary, and branded content. By leveraging machine learning-driven metadata processing, the system ensures that listeners receive highly relevant, real-time engagement options, regardless of whether they are listening live or consuming archived content.

[0502] Furthermore, the system’s scalability and adaptability allow broadcasters to tailor interactive experiences to specific audience demographics, regional markets, or content genres. For example, a sports broadcaster optionally enable real-time betting or game statistics integration, while a music station optionally facilitate merchandise purchases, concert bookings, or artist profile discovery.

[0503] In summary, by implementing AI-driven listener interactivity, real-time metadata synchronization, and commerce-enabled engagement, the system transforms traditional passive radio consumption into an interactive and highly personalized experience. Whether for live, scheduled, or archived broadcasts, the seamless integration of RadioTAG, RadioEPG, and bidirectional commerce workflows ensures that radio content remains interactive, contextually enriched, and commercially viable across multiple digital platforms.

[0504] Further optionally, the system architecture is designed to handle both live and archived content workflows efficiently, providing a scalable framework for metadata processing and listener engagement. The architecture comprises:

[0505] Live Radio Flow (Real-Time Processing)

a. **Hybrid Radio Metadata Retrieval** – The system retrieves metadata via the RadioDNS API, SPI, and other hybrid radio standards.

b. **AI-Driven Content Recognition** – Audio fingerprints and embedded metadata are processed in real time to generate enriched subject data.

c. **Content Synchronization at Workstation Platform** – Enriched metadata is dynamically linked to interactive subjects, providing real-time engagement opportunities.

d. **Listener Interaction Processing** – Listeners interact with content through a user device, triggering actions such as purchases, bookings, and social interactions.

5 **[0506] Pre-Programmed Radio Flow (Scheduled and Archived Content)**

a. **Preloading Metadata via RadioEPG** – Metadata is retrieved before scheduled airings, allowing AI-driven enrichment before content is broadcast.

b. **Batch Processing of Archived Audio** – Audio segments are scanned post-broadcast for subject tagging and metadata enrichment.

10 c. **Post-Broadcast Interaction Handling** – Listener interactions with archived content are processed, allowing users to engage with past broadcasts as they would with live content.

15 **[0507]** By integrating real-time metadata retrieval with AI-powered enrichment and commerce-enabled interactions, the system enhances user engagement and provides a robust, scalable framework for hybrid radio interaction.

20 **[0508]** For example, the live radio processing workflow begins when the system retrieves real-time metadata from hybrid radio metadata providers using the RadioDNS API, SPI (ETSI TS 102 818), and other hybrid radio standards. The metadata includes station identifiers, program schedules, live track information, presenter details, and any available interactive elements such as RadioTAG markers. This metadata is continuously streamed into the workstation platform, where it is indexed, structured, and linked to interactive content elements.

25 **[0509]** As the radio broadcast is processed, the system applies AI-driven content recognition, which extracts audio fingerprints from the live stream, identifies key segments, and matches them against a pre-indexed metadata repository. If embedded metadata such as audio watermarks, DAB service labels, or RadioEPG markers are present, they are immediately extracted and cross-referenced with the fingerprint-matched metadata to ensure synchronization accuracy. This dual-method approach ensures that content recognition remains highly reliable, even in the absence of directly embedded identifiers.

30 **[0510]** Once enriched metadata is generated, it is dynamically synchronized at the workstation platform and linked to interactive subjects. For example, if a radio segment discusses a product, an event, or an artist, the system automatically retrieves contextual links, e-commerce integrations, or event booking options. These elements are then transmitted via a bidirectional communication protocol to user devices, ensuring that listeners receive relevant engagement opportunities in real time.

35 **[0511]** Listener interactions are processed through an SDK-enabled interface, allowing users to tag moments in the broadcast, bookmark segments for later retrieval, share content on social media, or trigger commerce-enabled actions such as purchasing, booking, or subscribing. The system immediately processes these interactions, logs user engagement metrics, and ensures that broadcasters receive real-time feedback on audience participation. Additionally, the

metadata enrichment engine continuously updates based on listener interactions, refining future content recommendations and engagement opportunities.

5 [0512] Alternatively, for pre-programmed or archived content, the system follows a batch-processing workflow, where metadata is retrieved before a program airs via RadioEPG (ETSI TS 102 818). This allows the AI-driven workstation to pre-tag upcoming content with interactive elements, commerce links, and contextual subject associations, ensuring that when the broadcast occurs, listeners have access to pre-synchronized engagement options.

10 [0513] For post-broadcast engagement, the system scans archived audio and applies AI-based subject extraction, speech-to-text transcription, and contextual analysis to tag and categorize past content. Listeners accessing previously aired programs on on-demand platforms benefit from the same interactive functionality as live listeners, enabling them to re-engage with commerce opportunities, retrieve tagged moments, and access enriched metadata retroactively.

15 [0514] An advantage of this system architecture is its seamless integration of live and pre-programmed content workflows, ensuring that metadata enrichment, listener interaction, and commercial opportunities remain consistent across all radio formats. Unlike traditional one-way metadata tagging, this system leverages bidirectional communication, enabling real-time content adaptation, dynamic metadata updates, and audience-driven interaction management.

20 [0515] Beneficially, the scalable and modular design of this system allows it to be implemented across diverse radio distribution models, including terrestrial DAB+ broadcasting, hybrid internet radio, and on-demand streaming platforms. By employing AI-driven recognition models and automated metadata synchronization, the system minimizes broadcaster workload while maximizing audience engagement and interaction potential.

25 [0516] The real-time processing architecture ensures that metadata synchronization remains highly accurate, reducing issues such as content misalignment, lag in engagement options, or incorrect contextual tagging. Whether listeners engage with live, scheduled, or archived content, they receive consistent, enriched, and interactive experiences.

30 [0517] Furthermore, the system's flexible metadata framework allows for continuous updates and refinements, enabling broadcasters to adapt metadata strategies based on audience analytics, engagement trends, and emerging content preferences. Dynamic commercial insertion, commerce integration, and real-time interactivity ensure that radio broadcasting evolves beyond passive listening into an interactive, revenue-generating digital experience.

35 [0518] By integrating real-time metadata retrieval, AI-powered content enrichment, and commerce-enabled interaction processing, the system enhances user engagement, strengthens broadcaster interaction opportunities, and ensures that hybrid radio remains a relevant and interactive medium in the digital ecosystem. Whether deployed for live programming, scheduled broadcasts, or archived content retrieval, this system architecture establishes a future-proof foundation for scalable and intelligent radio interaction.

40 [0519] The present disclosure optionally incorporates machine learning techniques to enable real-time content recognition, metadata enrichment, and interactive listener engagement in live

and/or pre-programmed radio broadcasts as described above. Non-limiting example implementations of the aforementioned machine learning and AI models are described below.

[0520] For example, the AI-driven system processes, identifies, and synchronizes content, ensuring that audio streams are contextually enriched with relevant metadata and that listeners optionally engage with the broadcast seamlessly and interactively. The machine learning model(s) require high-quality, structured datasets to ensure accurate recognition, tagging, and enrichment of radio broadcasts. Data sources include audio streams retrieved from hybrid radio providers via DAB+, RTSP, HLS, and other broadcast protocols, metadata from RadioDNS, SPI, and EPG sources containing station information, program guides, and embedded metadata markers, AI-generated speech-to-text transcriptions allowing text-based metadata extraction for contextual enrichment, and historical user engagement data collected from RadioTAG events, commerce transactions, and content tagging trends.

[0521] Each dataset preferably undergoes preprocessing to enhance usability. The system transforms raw audio streams into structured feature representations such as spectrogram analysis, mel-frequency cepstral coefficients (MFCCs), and waveform-based embeddings for fingerprinting. Text-based metadata is normalized to remove duplicates, erroneous entries, and formatting inconsistencies to ensure structured, machine-readable data. Live audio signals are processed for noise reduction, amplitude normalization, and speech isolation, ensuring that the system maintains high accuracy even in noisy or complex broadcast environments. The system timestamps radio segments to map AI-generated insights to specific time points, ensuring that listener interactions align accurately with content. The minimum dataset size required for effective training varies depending on the task, with basic audio fingerprinting requiring at least 500,000 labelled audio segments and deep learning models for content classification necessitating millions of training samples, depending on the complexity of content differentiation. Alternatives include synthetic augmentation, where additional training data is created by modifying pitch, speed, or background noise levels to simulate varied listening conditions.

[0522] The machine learning models are optionally trained using a multi-stage approach that ensures robust recognition, efficient metadata mapping, and scalable performance. Initially, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are pretrained on publicly available datasets such as LibriSpeech, Mozilla Common Voice, and proprietary radio archives to develop a baseline understanding of speech patterns, music structures, and content variations. Training is performed using supervised learning for audio fingerprinting, where labelled datasets map specific sounds, words, or musical patterns to known metadata, and unsupervised learning for contextual extraction, where clustering algorithms group similar topics, voices, or content types, enabling real-time discovery of emerging topics or trends. Models are trained using backpropagation with Adam optimizer to minimize classification loss functions, batch normalization and dropout regularization to prevent overfitting, and an iterative training cycle with 50+ epochs to ensure convergence. Retraining occurs continuously, where new radio data is periodically incorporated to allow adaptation to changing speech patterns, accents, and new content types, with frequency determined by content variability and listener engagement metrics. The system supports alternative training paradigms such as fine-tuning smaller task-specific models or leveraging transformer-based architectures like Wav2Vec2 for improved speech modelling.

[0523] The system preferably integrates multiple specialized machine learning models to perform different tasks efficiently. An audio fingerprinting model based on CNN feature extraction generates compact hash representations of live and archived broadcasts, a speech-to-text model utilizing transformer architectures such as Wav2Vec2 or DeepSpeech performs real-time conversion of spoken words to structured metadata, and a contextual metadata mapping model uses a natural language processing (NLP) pipeline to associate extracted phrases with enriched metadata, commerce actions, or event recommendations. The inputs to these models include real-time audio streams in WAV, MP3, or FLAC format, metadata in JSON or XML format from RadioDNS, SPI, and EPG sources, and user interaction logs from RadioTAG events, purchases, and engagement metrics. The output consists of tagged content segments mapped to products, artists, or events, interactive engagement prompts such as "Buy Now" or "More Info," and aggregated user behavior insights for broadcaster analytics. The system is optimized for real-time inferencing, achieving prediction latencies under 200ms to ensure seamless user interactions.

[0524] The machine learning components are modular and integrate into the hybrid radio architecture via secure APIs and real-time data pipelines. The models operate within the workstation platform, processing audio continuously and synchronizing metadata with listener interactions. The system is compatible with DAB+, HLS, and RTSP streaming formats, enabling broadcasters to apply AI enrichment to live audio feeds. Archived broadcasts are batch-processed post-airing, allowing listeners to interact with enriched metadata retroactively. AI-driven recommendations, purchases, and content tagging enhance user experience and provide broadcasters with increased interaction opportunities.

[0525] The disclosed system comprises several technical advantages over conventional manual tagging and static metadata processing. The AI-powered framework eliminates the need for manual metadata tagging, reducing broadcaster workload while ensuring that content is enriched with dynamic and relevant metadata. The combination of audio fingerprinting with metadata retrieval ensures reliable content identification, even in low-data environments or when direct metadata embedding is unavailable. Unlike traditional radio broadcasting, which primarily relies on static metadata guides and non-interactive listener experiences, the present system enables continuous listener engagement through real-time interaction handling, ensuring that metadata remains adaptable to dynamic content. The models are hardware-agnostic, allowing deployment on cloud servers, edge devices, and embedded radio receivers, ensuring that the system remains scalable and adaptable across various broadcast ecosystems.

[0526] A real-time processing architecture ensures that metadata synchronization remains highly accurate, minimizing issues such as content misalignment, lag in engagement options, or incorrect contextual tagging. The system's flexible metadata framework supports continuous updates and refinements, allowing broadcasters to adapt metadata strategies based on audience analytics, engagement trends, and emerging content preferences. The integration of dynamic advertisement insertion, commerce integration, and real-time interactivity ensures that radio broadcasting evolves beyond passive listening into an interactive, revenue-generating digital experience. By integrating real-time metadata retrieval, AI-powered content enrichment, and commerce-enabled interaction processing, the system enhances user engagement, strengthens broadcaster interaction opportunities, and ensures that hybrid radio

remains a relevant and interactive medium in the digital ecosystem. Whether deployed for live programming, scheduled broadcasts, or archived content retrieval, this system architecture establishes a future-proof foundation for scalable and intelligent radio interaction.

5 [0527] Figure 25 shows an example computing environment for performing any of the methods described herein. Specifically, Figure 25 shows a block diagram of an embodiment of a computing system 2500 according to example embodiments of the present disclosure.

10 [0528] Computing system 2500 can be configured to perform any of the operations disclosed herein such as, for example, any of the processes or methods described herein. Computing system 2500 includes one or more computing devices 2502. Computing device 2502 of computing system 2500 comprises one or more processors 2504 and memory 2506. device(s) 2502 of computing system 2500 comprise one or more processors 2504 and memory 2506. One or more processors 2504 can be any general-purpose processor(s) configured to execute a set of instructions. For example, one or more processors 2504 can be one or more general-purpose processors, one or more field programmable gate array (FPGA), and/or one or more application specific integrated circuits (ASIC). In one embodiment, one or more processors 15 2504 include one processor. Alternatively, one or more processors 2504 include a plurality of processors that are operatively connected. One or more processors 2504 are communicatively coupled to memory 2506 via address bus 2508, control bus 2510, and data bus 2512. Memory 2506 can be a random-access memory (RAM), a read-only memory (ROM), a persistent storage device such as a hard drive, an erasable programmable read-only memory (EPROM), and/or the like. Computing device(s) 2502 further comprise input/output (I/O) interface 2514 20 communicatively coupled to address bus 2508, control bus 2510, and data bus 2512.

[0529] Memory 2506 can store information that can be accessed by one or more processors 2504. For example, memory 2506 (e.g., one or more non-transitory computer-readable storage 25 mediums, memory devices) can include computer-readable instructions (not shown) that can be executed by one or more processors 2504. Alternatively, or additionally, memory 2506 can be a combination of information storage and the option to process data in-memory without having to read out the data first, and let the processor 2504 perform operations on the memory data. For example, in memory computing is optionally used for CTM implementation, e.g. for 30 “weighting” of clauses, where the weights from a trained model are stored in memory and the weighting is automatically performed when a clause evaluates to 1.

[0530] The computer-readable instructions can be software written in any suitable programming language or can be implemented in hardware. Additionally, or alternatively, the computer-readable instructions can be executed in logically and/or virtually separate threads 35 on one or more processors 2504. For example, memory 2506 can store instructions (not shown) that when executed by one or more processors 2504 cause one or more processors 2504 to perform operations such as any of the operations and functions for which computing system 2500 is configured, as described herein. In addition, or alternatively, memory 2506 can store data (not shown) that can be obtained, received, accessed, written, manipulated, created, 40 and/or stored. The data can include, for instance, the data and/or information described herein in relation to Figures 1 to 11. In some implementations, computing device(s) 2502 can obtain from and/or store data in one or more memory device(s) that are remote from the computing system 2500.

[0531] Computing system 2500 further comprises storage unit 2516, network interface 2518, input controller 2520, and output controller 2522. Storage unit 2516, network interface 2518, input controller 2520, and output controller 2522 are communicatively coupled to central control unit or computing devices 2502 via I/O interface 2514.

5 [0532] Storage unit 2516 is a computer readable medium, preferably a non-transitory computer readable medium, comprising one or more programs, the one or more programs comprising instructions which when executed by one or more processors 2504 cause computing system 2500 to perform the method steps of the present disclosure. Alternatively, storage unit 2516 is a transitory computer readable medium. Storage unit 2516 can be a
10 persistent storage device such as a hard drive, a cloud storage device, or any other appropriate storage device.

[0533] Network interface 2518 can be a Wi-Fi module, a network interface card, a Bluetooth module, and/or any other suitable wired or wireless communication device. In an embodiment, network interface 2518 is configured to connect to a network such as a local area network
15 (LAN), or a wide area network (WAN), the Internet, or an intranet.

[0534] Figure 25 illustrates one example computer system 2500 that can be used to implement the present disclosure. Other computing systems can be used as well. Computing tasks discussed herein as being performed at and/or by one or more functional unit(s) can instead be performed remote from the respective system, or vice versa. Such configurations can be
20 implemented without deviating from the scope of the present disclosure. The use of computer-based systems allows for a great variety of possible configurations, combinations, and divisions of tasks and functionality between and among components. Computer-implemented operations can be performed on a single component or across multiple components. Computer-implemented tasks and/or operations can be performed sequentially or in parallel.
25 Data and instructions can be stored in a single memory device or across multiple memory devices.

[0535] Regarding the above disclosure, references to items in the singular should be understood to include items in the plural, and vice versa, unless explicitly stated otherwise or clear from the context. Additionally, grammatical conjunctions are intended to express any and
30 all disjunctive and conjunctive combinations of conjoined clauses, sentences, words, and the like, unless otherwise stated or clear from the context. Thus, the term “or” should generally be understood to mean “and/or” and so forth. The use of any and all examples, or exemplary language (“e.g.,” “such as,” “including,” or the like) provided herein, is intended merely to better illuminate the embodiments and does not pose a limitation on the scope of the
35 embodiments or the claims.

[0536] Methods described herein may relate to a computer storage product with a non-transitory computer-readable medium (also can be referred to as a non-transitory processor-readable medium) having instructions or computer code thereon for performing various computer-implemented operations, such that the methods are performed. The computer-readable medium (or processor-readable medium) is non-transitory in the sense that it does
40 not include transitory propagating signals per se (e.g., a propagating electromagnetic wave carrying information on a transmission medium such as space or a cable). The media and computer code (also can be referred to as code) may be those designed and constructed for

the specific purpose or purposes. Examples of non-transitory computer-readable media include, but are not limited to, magnetic storage media such as hard disks, floppy disks, and magnetic tape, optical storage media such as Compact Disc/Digital Video Discs (CD/DVDs), Compact Disc-Read Only Memories (CD-ROMs), and holographic devices; magneto-optical storage media such as optical disks; carrier wave signal processing modules; and hardware devices that are specially configured to store and execute program code, such as Application-Specific Integrated Circuits (ASICs), Programmable Logic Devices (PLDs), Read-Only Memory (ROM) and Random-Access Memory (RAM) devices. Other embodiments described herein relate to a transitory computer program product, which can include, for example, the instructions and/or computer code discussed herein.

[0537] Some embodiments and/or methods described herein can be performed by software (executed on hardware), hardware, or a combination thereof. Hardware modules include, for example, a general-purpose processor, a field programmable gate array (FPGA), and/or an application specific integrated circuit (ASIC). Software modules (executed on hardware) can be expressed in a variety of software languages (e.g., computer code), including C, C++, Java, Ruby, Visual Basic, Python, and/or other object-oriented, procedural, or other programming language and development tools. Examples of computer code include, but are not limited to, micro-code or micro-instructions, machine instructions, such as produced by a compiler, code used to produce a web service, and files containing higher-level instructions that are executed by a computer using an interpreter. For example, embodiments can be implemented using imperative programming languages (e.g., C, Fortran, etc.), functional programming languages (Haskell, Erlang, etc.), logical programming languages (e.g., Prolog), object-oriented programming languages (e.g., Java, C++, etc.) or other suitable programming languages and/or development tools. Additional examples of computer code include, but are not limited to, control signals, encrypted code, and compressed code. Optionally, the embodiments and/or methods described herein are implemented using an operating system such as Robot Operating System (ROS).

[0538] The following statements encompass example embodiments of the systems and method described herein, and do not define the scope of the invention, which is instead defined in the appended claims.

Statements of invention:

1. A method for media content identification and content information retrieval at a user device, the method comprising:

monitoring, using a microphone of the user device, a portion of an audio track emitting from a speaker, wherein the audio track is associated with a media content;
detecting whether there is an audio tag embedded within the portion of the audio track, wherein the audio tag is embedded at frequencies detectable by the microphone and less detectable to a user of the user device;

if an audio tag is detected:

extracting identification data from the audio tag, wherein the identification data is associated with a known media content at a known timepoint;

if an audio tag is not detected:

- receiving identification data based on a best match for an audio fingerprint from a plurality of audio fingerprints, wherein the audio fingerprint comprises one or more audio features extracted from the portion of the audio track;
- identifying the media content based on the known media content of the identification data; and
- retrieving content information associated with the identification data.
2. The method of statement 1, wherein retrieving content information comprises:
- transmitting, to a server device, an information request comprising the identification data; and
- obtaining, from the server device, information associated with one or more subjects relevant to the known media content at the known timepoint, wherein each subject of the one or more subjects is linked to the known timepoint.
3. The method of statement 2, wherein the information request is transmitted based on a user interaction with the user device.
4. The method of any preceding statement, wherein the speaker is associated with a media device independent of the user device.
5. The method of any preceding statement, further comprising:
- identifying a timepoint of the media content based on the known timepoint of the identification data; and
- displaying content information on the user device based on the timepoint, such that displayed content information is relevant to the media content at the known timepoint.
6. The method of any preceding statement, wherein the microphone is switched on or off at periodic intervals while monitoring the portion of the audio track.
7. The method of any preceding statement, wherein the audio tag is embedded using frequency-shift keying modulation at defined intervals.
8. The method of statement 14, wherein the defined intervals are adjusted after the media content is identified.
9. The method of any preceding statement, wherein the known timepoint is a predetermined time interval.
10. The method of any preceding statement, wherein the plurality of audio fingerprints comprises:
- audio fingerprints associated with a plurality of portions of an audio track at a plurality of time points; or
- audio fingerprints associated with one or more portions of a plurality of audio tracks.
11. The method of any preceding statement, wherein receiving identification data based on the best match comprises:
- transmitting the audio fingerprint or the portion of the audio track to a server device, wherein the audio fingerprint is generated by extracting the one or more audio features from the portion of the audio track; and
- obtaining, from the server device, identification data associated with the best match, wherein the server device is configured to compare the audio fingerprint against the plurality of audio fingerprints stored in a database accessible to the server device, thereby determining the best match.

12. The method of any preceding statement, wherein detecting whether there is an audio tag embedded within the portion of the audio track comprises application of one or more correction algorithms to compensate for noise, interference, or other audio degradation factors.

5 13. The method of any preceding statement, wherein detecting whether there is an audio tag is limited to a predefined time frame.

14. A computer-readable medium storing instructions for performing the method of any of statements 1-13.

15. A user device comprising:

10 a microphone;
a user interface; and
a memory storing instructions that, when executed, cause the user device to perform the method of any of statements 1-13.

16. A method for audio track identification and content information transmission at a server device, the method comprising:

15 obtaining, from a user device, either:
a first portion of an audio track; or
one or more audio features extracted from the first portion;
if one or more audio features are not obtained from the user device, extracting the one or more audio features from the first portion;
20 generating a first audio fingerprint comprising the one or more audio features;
comparing the first audio fingerprint to a plurality of audio fingerprints, wherein each audio fingerprint of the plurality of audio fingerprints is associated with identification data, and the identification data is associated with a known audio track at a known timepoint;
25 identifying a best match, wherein the best match is an audio fingerprint of the plurality of audio fingerprints most similar to the first audio fingerprint;
transmitting, to the user device, identification data based on the best match; and
if a request is received from the user device, transmitting content information associated with the identification data to the user device.

30 17. The method of statement 16, further comprising:

obtaining, from the user device, a second portion of the audio track or one or more audio features extracted from the second portion;
generating a second audio fingerprint;
identifying an updated best match, wherein the updated best match is an audio fingerprint of the plurality of audio fingerprints most similar to second audio fingerprint;
35 and
transmitting, to the user device, updated content information, wherein the updated content information is associated with identification data based on the updated best match.

40 18. The method of any of statements 16-17, wherein the plurality of audio fingerprints is stored in a database.

19. The method of statement 18, wherein the database is configured to retrieve the identification data based on the first audio fingerprint and retrieve the content information based on the identification data.

20. The method of any of statements 16-19, wherein the request is associated with a user interaction with the user device.

21. The method of any of statements 16-20, wherein extracting one or more audio features comprises analysing an audio signal of the audio track to identify or quantify selected audio characteristics.

22. The method of any of statements 16-21, wherein generating a first audio fingerprint comprises hashing the first portion by encoding the one or more features extracted from the first portion.

23. The method of any of statements 16-22, wherein comparing the first audio fingerprint to the plurality of audio fingerprints comprises performing a nearest neighbour search.

24. The method of statement 23, wherein a nearest neighbour search is performed using a ball tree algorithm.

25. The method of any of statements 16-24, wherein identifying a best match comprises identifying a plurality of best matches and transmitting identification data associated with each best match to the user device.

26. The method of any of statements 16-25, wherein the known timepoint is a predetermined time interval.

27. The method of any of statements 16-26, wherein content information comprises information associated with one or more subjects relevant to the known audio track at the known time point.

28. The method of statement 27, wherein the known audio track is associated with a known media content depicting or otherwise associated with the one or more subjects.

29. A computer-readable medium storing instructions for performing the method of any of statements 16-28.

30. A server device comprising:

a processor; and

a memory storing instructions that, when executed, cause the processor to perform the method of any of statements 16-28.

31. A method for media content processing, the method comprising:

obtaining a known media content comprising a known audio track;

generating a timeline for the known media content, wherein the timeline comprises a plurality of timepoints;

for a first timepoint of the timeline:

extracting one or more audio features from the known audio track at the first timepoint for generating a first audio fingerprint comprising the one or more audio features;

linking identification data to the first audio fingerprint, wherein the identification data is associated with the known media content at the first timepoint;

embedding a first audio tag within the known audio track, wherein the first audio tag comprises the identification data, thereby generating an embedded audio track;

extracting one or more subjects from the known media content;

collating information associated with each subject of the one or more subjects, thereby generating content information associated with the identification data;

and

storing the first audio fingerprint, the identification data, the embedded audio track, and the content information, wherein the identification data is retrievable based on the first audio fingerprint or the embedded audio track, and wherein the content information is retrievable based on the identification data.

5 32. The method of statement 31, further comprising embedding a second audio tag within the embedded audio track, wherein the second audio tag comprises identification data associated with the known media content at a second timepoint of the plurality of timepoints.

33. The method of any of statements 31-32, wherein the first audio fingerprint, the embedded audio track, and the content information is stored in a database, the database configured for:

10 retrieval of the identification data based on the first audio fingerprint or the embedded audio track; or

retrieval of the content information based on the identification data.

34. The method of statement 33, wherein storing the first audio fingerprint, the identification data, the embedded audio track, and the content information is performed at a server device
15 configured to access the database.

35. The method of statement 34, wherein obtaining a known media content comprises importing the known media content from the server device at a workstation device.

36. The method of any of statements 31-35, wherein the known media content comprises video content, interactive content, streamed content, broadcast content, or user-generated content.

20 37. The method of any of statements 31-36, wherein each timepoint of the plurality of timepoints is a predetermined time interval.

38. The method of statement 37, wherein a total duration of the plurality of timepoints is collectively at least a duration of the known media content.

25 39. The method of any of statements 31-38, wherein the known media content comprises a plurality of known audio tracks.

40. The method of any of statements 31-39, wherein extracting one or more audio features comprises analysing an audio signal of the known audio track to identify or quantify selected audio characteristics.

30 41. The method of statement 40, wherein analysing the audio signal comprises performing any of:

a feature extraction technique comprising Fourier transforms, mel-frequency cepstral coefficients, chroma feature, spectral centroid, zero-crossing rate, root mean square energy, or other standard technique;

statistical analysis, temporal analysis, feature scaling, or visualisation analysis; or

35 one or more machine learning techniques.

42. The method of any of statements 31-41, wherein the audio tag is embedded using frequency-shift keying modulation at defined intervals.

43. The method of statement 42, wherein the audio tag is embedded at frequencies detectable by the microphone and less detectable to a user of the workstation device.

40 44. A computer-readable medium storing instructions for performing the method of any of statements 31-43.

45. A workstation device comprising:

a processor; and

a memory storing instructions that, when executed, cause the processor to perform the
45 method of any of statements 31-43.

46. A system for media content interaction, the system comprising a workstation device, a server device, and a user device, wherein the system is configured to:

process a known media content at the workstation device, wherein processing the known media content comprises:

5 obtaining the known media content comprising a known audio track;
generating a timeline from the known media content, wherein the timeline comprises a plurality of timepoints;
for each timepoint of the timeline:

10 generating an embedded audio track by embedding an audio tag comprising identification data within the known audio track, wherein the audio tag is embedded at frequencies detectable by the microphone and less detectable to a user of the user device, and wherein the identification data is associated with the known media content at the timepoint; and

15 generating content information associated with the identification data by collating information associated with one or more subject extracted from the known media content at the timepoint; and

storing the embedded audio track and the content information associated with each timepoint of the timeline;

20 interact with a media content at the user device, wherein interacting with the media content comprises:

monitoring a portion of an unknown audio track using a microphone of the user device, wherein the unknown audio track is associated with the media content emitting from a speaker;

25 identifying the media content by acquiring identification data, wherein:

if an audio tag is detected within the portion of the unknown audio track, identification data is extracted from the audio tag, wherein the identification data is associated with the known media content at a known timepoint; or

30 if an audio tag is not detected within the portion of the unknown audio track, identification data is received from the server device; and

initiating, based on a user interaction with the user device, a request for content information from the server device; and

35 transmit content information associated with the identification data from the server device to the user device, wherein transmitting content information comprises:

receiving the request for content information from the user device, wherein the request comprises the identification data;

retrieving stored content information associated with the identification data; and outputting the content information to the user device.

40 47. The system of statement 46 wherein processing the known media content at the workstation device further comprises:

generating an audio fingerprint for each timepoint of the timeline, thereby generating a plurality of audio fingerprints, wherein the audio fingerprint comprises one or more audio features extracted from the known audio track at the timepoint; and

linking identification data to each audio fingerprint of the plurality of audio fingerprints based on the plurality of timepoints of the timeline.

48. The system of statement 47, wherein receiving identification data from the server device comprises, at the server device:

- 5 obtaining one or more audio features extracted from the portion of the unknown audio track monitored by the user device;
generating an unknown audio fingerprint comprising the one or more audio features;
comparing the unknown audio fingerprint to the plurality of audio fingerprints identifying a best match, wherein the best match is an audio fingerprint of the plurality of audio fingerprints most similar to the unknown audio fingerprint; and
10 transmitting identification data associated with the best match to the user device.

49. The system of any of statements 46-48, wherein the server device or the workstation device is communicatively coupled to a database, and wherein stored content information is retrieved from the database.

15 50. The system of statement 49, wherein the database is configured to store the embedded audio track and the content information associated with each timepoint of the timeline.

51. The system of any of statements 46-50, wherein the workstation device comprises a processor and a memory storing instructions that, when executed, cause the processor to process known media content.

20 52. The system of any of statements 46-51, wherein the user device comprises a user interface configured to obtain a user interaction with the user device, a processor, and a memory storing instructions that, when executed, cause the processor to interact with a media content.

53. The system of any of statements 46-52, wherein the system is further configured to:

- 25 embed, at the workstation device, a second audio tag within a second known audio track at the workstation device, wherein the second audio tag comprises identification data associated with the second known audio track, thereby generating a second embedded audio track; and
storing the second embedded audio track.

54. The system of statement 53, wherein the system is further configured to:

- 30 monitor, at the user device, a portion of a second unknown audio track using the microphone;
extract the identification data from the second audio tag detected within the portion of the second unknown audio track, thereby identifying the second known audio track as the second known audio track;
35 request content information associated with the second known audio track by transmitting the identification data to the server device; and
receive, at the user device, content information associated with the second known audio track.

40 55. The system of any of statements 46-54, wherein monitoring a portion of an unknown audio track using a microphone of the user device comprises:

- monitoring an audio signal, wherein the audio signal comprises the portion of an unknown audio track; and
processing the audio signal to improve detection of the audio tag.

45 56. The system of statement 125, wherein the audio signal further comprises a portion of another unknown audio track and monitoring the audio signal comprises:

detecting a plurality of audio tags;
extracting identification data for each of the plurality of audio tags, associated with a plurality of known media content; and
receiving a user interaction selecting the known media content from the plurality of known media content.

57. A method for media content interaction, the method performed at a workstation device, a server device, and a user device, wherein the method comprises:

processing a known media content at the workstation device, wherein processing the known media content comprises:

obtaining the known media content comprising a known audio track;

generating a timeline from the known media content, wherein the timeline comprises a plurality of timepoints;

for each timepoint of the timeline:

generating an embedded audio track by embedding an audio tag comprising the identification data within the known audio track, wherein the audio tag is embedded at frequencies detectable by the microphone and less detectable to a user of the user device; and

generating content information associated with the identification data by collating information associated with one or more subject extracted from the known media content at the timepoint; and

storing the embedded audio track and the content information associated with each timepoint of the timeline;

interacting with a media content at the user device, wherein interacting with the media content comprises:

monitoring a portion of an unknown audio track using a microphone of the user device, wherein the unknown audio track is associated with the media content emitting from a speaker;

identifying the media content by acquiring identification data, wherein:

if an audio tag is detected within the portion of the unknown audio track, identification data is extracted from the audio tag, wherein the identification data is associated with the known media content at a known timepoint; or

if an audio tag is not detected within the portion of the unknown audio track, identification data is received from the server device; and

initiating, based on a user interaction with the user device, a request for content information from the server device; and

transmitting content information associated with the identification data from the server device to the user device, wherein transmitting content information comprises:

receiving the request for content information from the user device, wherein the request comprises the identification data;

using the identification data, retrieving stored content information associated with the identification data; and

outputting the content information to the user device.

58. The method of statement 57, wherein processing the known media content at the workstation device further comprises:

generating an audio fingerprint for each timepoint of the timeline, thereby generating a plurality of audio fingerprints, wherein the audio fingerprint comprises one or more audio features extracted from the known audio track at the timepoint; and linking identification data to each audio fingerprint of the plurality of audio fingerprints based on the plurality of timepoints of the timeline.

59. The method of statement 58, wherein receiving identification data from the server device comprises, at the server device:

obtaining one or more audio features extracted from the portion of the unknown audio track monitored by the user device;
generating an unknown audio fingerprint comprising the one or more audio features;
comparing the unknown audio fingerprint to the plurality of audio fingerprints identifying a best match, wherein the best match is an audio fingerprint of the plurality of audio fingerprints most similar to the unknown audio fingerprint; and
transmitting identification data associated with the best match to the user device.

60. A computer-readable medium storing instructions for performing the method of any of statements 57-59.

61. A method for preparing content interaction at a workstation platform, the method comprising:

importing a media content to the workstation platform, wherein the media content is associated with a plurality of subjects and a media content identifier;
generating a timeline associated with the media content, wherein the timeline comprises a plurality of timepoints;
obtaining one or more subject files at the workstation platform, wherein each subject file is associated with a subject of the plurality of subjects and comprises a visual depiction of the subject and a subject text;
assigning each subject file to at least one timepoint on the timeline;
outputting an enriched media file, wherein the enriched media file comprises the timeline, the one or more subject files associated with the timeline, and the media content identifier.

62. The method of statement 61, wherein obtaining one or more subject files comprises using an artificial intelligence algorithm to:

combine the visual depiction and the subject text;
retrieve or generate the visual depiction or the subject text; or
provide relevant data associated with the subject.

63. The method of statement 62, further comprising editing, removing, adding to, or correcting the relevant data, the visual depiction, or the subject text.

64. The method of statement 61, wherein the media content comprises an audio track and the method further comprises:

embedding an audio tag at a first timepoint within the audio track, wherein the audio tag comprises identification data associated with the media content identifier and the first timepoint, thereby generating an embedded audio track; and
the enriched media file further comprises the embedded audio track.

65. The method of statement 64, wherein any subject files assigned to the first timepoint are associated with the audio tag.

66. The method of statement 65, wherein at least one subject file assigned to the first timepoint is provided to a user device based on the audio tag being detected at the user device.

67. The method of any of statements 61-66, wherein outputting the enriched media file comprises storing the enriched media file in a database.

5 68. The method of any of statements 61-67, wherein outputting the enriched media file comprises downloading the enriched media file from the workstation platform.

69. The method of statement 68, wherein importing the media content to the workstation platform comprises uploading a previously downloaded enriched media file.

10 70. The method of any of statements 61-69, wherein the media content identifier is unique for each imported media content.

71. The method of any of statements 67-70, wherein the media content is associated with a set of media contents, and wherein the set of media contents is associated with a group identifier unique to each set of media contents.

15 72. The method of statement 71, wherein the enriched media file further comprises the group identifier.

73. The method of any of statements 61-72, wherein the one or more subjects comprises any of people, locations, products, services, or other item identifiable within the media content.

74. A computer-readable medium storing instructions for performing the method of any of statements 61-73.

20 75. A system comprising:

a processor; and

a memory storing instructions that, when executed, cause the processor to perform the method of any of statements 61-73.

25 76. A method for content interaction at a user platform, the method comprising:

monitoring an audio signal, wherein the audio signal comprises one or more audio tags embedded within the audio signal;

obtaining a selected audio tag from the audio signal, wherein the selected audio tag is used to identify a media content;

30 extracting identification data from the selected audio tag, wherein the identification data is associated with a media content identifier and a first timepoint;

if a lock-on indicator is not received at the user platform:

obtaining a request indicator for a first subject file associated with the media content identifier at the first timepoint;

35 transmitting the identification data to a server device and receiving, from the server device, the first subject file, wherein the first subject file comprises a first visual depiction and a first subject text;

initiating display of the first visual depiction or the first subject text from the first subject file; and

continuing to monitor the audio signal;

40 if a lock-on indicator is received at the user platform:

reducing monitoring of the audio signal;

transmitting the identification data to the server device, thereby requesting a plurality of subject files associated with the media content identifier at a plurality of timepoints, wherein the plurality of subject files comprises the first subject file;

45

receiving, from the server device, the plurality of subject files;
initiating display of the first visual depiction or the first subject text from the first subject file; and
when a second timepoint is reached, initiating display of a second visual depiction or a second subject text from a second subject file of the plurality of subject files, wherein the second subject file is associated with the second timepoint.

77. The method of statement 76, wherein obtaining a selected audio tag comprises:

extracting a plurality of audio tags from one or more audio signals; and

determining the selected audio tag from the plurality of audio tags based on a first user interaction with the user platform.

78. The method of any of statements 76-77, wherein the audio signal is associated with one or more audio tracks, each audio track associated with a media content.

79. The method of any of statements 76-78, wherein the media content comprises video or radio content.

80. The method of any of statements 76-79, wherein continuing to monitor the audio signal comprises:

monitoring the audio signal, wherein the audio signal comprises a new audio tag;

obtaining a new audio tag from the audio signal;

extracting new identification data from the new audio tag, wherein the new identification data is associated with the media content identifier and a second timepoint or a new media content identifier.

81. The method of any of statements 76-80, wherein the lock-on indicator is a second user interaction with the user platform.

82. The method of any of statements 76-81, wherein the request indicator for the first subject file is a third user interaction with the user platform.

83. The method of any of statements 76-82, wherein reducing monitoring of the audio signal comprises ceasing monitoring of the audio signal for a predetermined time period.

84. The method of any of statements 76-83, wherein the first subject file is associated with a plurality of subjects, and wherein:

each subject of the plurality of subjects is associated with a classification;

each subject of the plurality of subjects is associated with a visual depiction and a subject text, such that the first subject file comprises a plurality of visual depictions and a plurality of subject texts; and

the first visual depiction and the first subject text is selected from the plurality of visual depictions and the plurality of subject texts based on the classification of each subject.

85. The method of any of statements 76-84, further comprising, if a lock-on indicator is received at the user platform:

receiving a monitoring indicator, wherein the monitoring indicator is indicative of a time difference between reaching the second timepoint and the second visual depiction or the second subject text being displayed exceeding a threshold; and
resuming monitoring of the audio signal.

86. The method of statement 85, wherein the monitoring indicator is received when:

5 a user of the user platform leaves an area of transmission, the area of transmission comprising a geographical location associated with transmission, emission, broadcast, or streaming of the audio signal or the media content; or
the audio signal or the media content is paused, accelerated, decelerated, or otherwise interrupted or interfered with compared to a chronological continuation of the audio signal or the media content.

87. The method of any of statements 76-86, further comprising:

10 receiving a fourth user interaction with the user platform indicative of a request for further information associated with a displayed visual depiction or subject text; and
initiating display of relevant information associated with the displayed visual depiction or subject text.

88. The method of any of statements 76-87, further comprising:

receiving a fifth user interaction with the user platform associated with reviewing, sharing, or otherwise interacting with components of the first subject file.

15 89. A computer-readable medium storing instructions for performing the method of any of statements 76-88.

90. A system comprising:

20 a processor; and
a memory storing instructions that, when executed, cause the processor to perform the method of any of statements 76-88.

91. A method for secure communication and output synchronisation with a transmission application at a user application, the method comprising:

25 scanning for content by broadcasting a first request using a bidirectional communication protocol;
detecting, using a microphone controllable from the user application, a first sound code, wherein:

the first sound code is output from a speaker controllable from the transmission application; and

30 the first sound code is configured to be detectable by the microphone and less detectable to a user of the user application;

converting the first sound code into an identification code;

transmitting, using the bidirectional communication protocol, the identification code for authentication at the transmission application;

35 establishing a bidirectional connection for live data exchange with the transmission application; and

synchronising, using the bidirectional connection, a first output of the user application with a second output of the transmission application.

92. The method of statement 91, wherein:

40 the bidirectional communication protocol comprises WebSockets; and

the bidirectional connection comprises a full-duplex connection.

93. The method of any of statements 90 to 92, further comprising:

sustaining the bidirectional connection in response to detecting a second sound code within a predetermined interval after detecting the first sound code; or

terminating or otherwise closing the bidirectional connection in response to:

45 a user interaction with the user application; or

a location associated with the user application exceeding a predetermined threshold.

94. The method of any of statements 90 to 93, wherein the first sound code is an ultrasound or near-ultrasound code.

5 95. The method of any of statements 90 to 94, wherein:

the identification code comprises alphanumeric data; and

converting the first sound code into the identification code comprises decoding the first sound code.

10 96. The method of any of statements 90 to 95, wherein detecting the first sound code is in response to the transmission application:

receiving an identifier associated with the first request from a server;

converting the identifier into the first sound code; and

initiating output of the first sound code from the speaker.

97. The method of any of statements 90 to 97, wherein synchronising comprises:

15 identifying a current timestamp associated with the first output;

identifying a message timestamp associated with the second output sent from the transmission application; and

reducing a difference between the current timestamp and the message timestamp, thereby improving synchronicity between the first output and the second output.

20 98. A method for secure communication and output synchronisation with a user application at a transmission application, the method comprising:

receiving, using a bidirectional communication protocol, a first request associated with the user application;

25 transmitting a pairing request to a server, wherein the pairing request comprises data associated with the first request;

if the server identifies the pairing request, receiving an identifier from the server;

converting the identifier into a first sound code for emission from a speaker, wherein the speaker is controllable from the transmission application;

30 receiving, using the bidirectional communication protocol, an identification code from the user application;

authenticating the identification code based on matching the identifier and the identification code; and

if the identification code is authenticated:

35 performing a handshake, thereby establishing a bidirectional connection for live data exchange with the user application; and

synchronising, using the bidirectional connection, a first output of the user application with a second output of the transmission application.

99. The method of statement 98, wherein:

40 the bidirectional communication protocol comprises WebSockets; and

the bidirectional connection comprises a full-duplex connection.

100. The method of statement 98 or 99, wherein synchronisation comprises:

receiving, using the bidirectional connection, a message timestamp associated with the first output;

determining a current timestamp associated with the second output; and

reducing a difference between the current timestamp and the message timestamp, thereby improving synchronicity between the first output and the second output.

101. A computer-readable medium storing instructions for performing the method of any of statements 1-100.

5 102. A system for secure communication and output synchronisation between a user device and a transmission device, the system comprising:

a user device comprising a microphone and a user interface, wherein the user device is running a user application configured to perform the method of any of statements 1-7; and

10 a transmission device communicatively connected to a speaker and a content interface, wherein the transmission device is running a transmission application configured to perform the method of any of statements 98-100.

103. A Software Development Kit (SDK) for secure communication and output synchronisation, the SDK comprising:

15 one or more communication modules configured to:

send data from a transmission device to a user device using a bidirectional protocol;

receive data from a user device to the transmission device using the bidirectional protocol;

20 establish a bidirectional connection between the transmission device and the user device;

transmit data to a server;

receive data from the server; and

transmit instructions to a speaker;

25 one or more authentication modules configured to:

generate a pairing request associated with data received from the user device;

generating a first sound code by encoding an identifier received from the server; and

compare the identification code to the identifier received from the server; and

30 one or more interaction modules configured to:

identify a current timestamp associated with an output of the transmission device

calculate a difference between a message timestamp received from the user device and the current timestamp; and

35 initiate adjustment of the output of the transmission device to minimise the difference.

104. The SDK of statement 103, wherein:

the one or more communication modules are configured to establish a secure WebSocket connection (WSS) using a TLS/SSL protocol; and/or

40 generating the first sound code further comprises encoding the current timestamp with added redundancy to facilitate error detection and correction during transmission, emission, playback, or decoding.

105. A method for media content processing using hashing at a workstation device, the method comprising:

segmenting a known media content into a plurality of media segments by dividing the known media content into media segments of a predetermined duration;
hashing each media segment of the plurality of media segments, thereby generating a plurality of segment hashes, each segment hash associated with the known media
5 content at a time interval;
arranging the plurality of segment hashes on one or more lower hierarchical levels of a hierarchical hash structure;
wherein the hierarchical hash structure is configured by combining hash values from the one or more lower hierarchical levels to generate hash values of higher
10 hierarchical levels, culminating in a root hash associated with the known media content over all time intervals; and
associating the plurality of segment hashes with data comprising one or more context tags for differentiating between known media contents.

106. The method of statement 105, further comprising embedding an audio tag within the known media content, wherein the audio tag is detectable by a microphone and less
15 detectable to a user.

107. The method of statement 105 or 106, wherein the hierarchical hash structure is stored in a blockchain or distributed ledger for secure storage and verification of the root hash.

108. The method of any of statements 105 to 107, wherein the hierarchical hash structure is a
20 Merkle tree.

109. The method of any of statements 105 to 108, wherein the plurality of segmented hashes are cryptographic hashes, and each media segment is hashed using cryptographic algorithms, e.g. SHA-256.

110. The method of any of statements 105 to 109, further comprising implementing Reed-Solomon error correction for recovering from data loss during transmission.
25

111. The method of any of statements 105 to 110, wherein the known media content comprises both audio and visual data, such as a video.

112. The method of any of statements 105 to 111, wherein the predetermined duration is a 1 second interval.

30 113. The method of any of statements 105 to 112, wherein combining hash values from the one or more lower hierarchical levels comprises concatenating segmented hashes, thereby generating parent hashes, and the root hash is generated by concatenating parent hashes.

114. The method of any of statements 105 to 113, wherein the data comprises content information associated with the known media.

35 115. The method of statement 114, wherein a segment hash of the plurality of segment hashes is used to retrieve specific content information associated with the media segment corresponding to the segment hash.

116. The method of statement 114 or 115, wherein the data is pre-stored at a database.

40 117. The method of statement 116, further comprising using an artificial intelligence system to:

recognise subjects within the known media content; and

automatically associated recognised subjects with the pre-stored data.

118. A method for maintained media content synchronisation at a user device, the method comprising:

- receiving a root hash of a hierarchical hash structure associated with a media content,
wherein higher hierarchical levels of the hierarchical hash structure comprise
combined hash values from lower hierarchical levels, culminating in the root hash;
retrieving data from a database associated with the root hash, wherein the data
comprises a contextual tag for differentiating between known media contents and
content information for media content interaction;
monitoring the media content at a first timestamp for a predetermined duration,
thereby obtaining a first media segment;
verifying synchronisation by computing a hash of the first media segment and
comparing the hash with the hierarchical hash structure;
if synchronisation is verified, using the hash to retrieve and display content information
associated with the first timestamp;
if synchronisation is not verified, performing any of:
adjusting the first timestamp for resynchronising and displaying content
information associated with the adjusted first timestamp; or
based on one or more adjacent media segments, using a trained machine
learning model to generate a predicted hash for retrieving and displaying
content information associated with a predicted timestamp;
monitoring the media content at a second timestamp for the predetermined duration,
the second timestamp within the predetermined duration from the first timestamp,
thereby obtaining a second media segment overlapping with the first media segment;
and
computing a hash of the second media segment, thereby re-hashing overlapping
media segments for maintaining synchronisation.
119. The method of statement 118, further comprising, if synchronisation is not verified:
monitoring the media content to detect an audio tag, wherein the audio tag is an
ultrasound signal embedded within the media content; and
extracting content information associated with the audio tag.
120. The method of statement 119, further comprising switching between synchronising
based on detected audio tags and synchronising based on hashing based on one or more
synchronisation performance metrics.
121. The method of any of statements 118 to 120, wherein the media content comprises
video content.
122. The method of any of statements 118 to 121, wherein the method is performed by a user
application operating or otherwise installed on the user device.
123. The method of any of statements 122, wherein the user application utilises end-to-end
encryption of hashes and data communication between the user device and the database.
124. The method of any of statements 123, wherein the user application is configured to
enrich content information based on automated subject recognition.
125. The method of any of statements 118 to 124, wherein the hierarchical hash structure is a
Merkle tree.
126. The method of any of statements 118 to 125, wherein the root hash is stored using
blockchain technology.

5 [0539] Any of the above statements can be combined and the skilled person would understand such examples do not limit the potential embodiments of the present disclosure. Although one or more concepts have been explained in relation to the above embodiment, it is to be understood that many other possible modifications and variations can be made without departing from the spirit and scope.

10 [0540] The skilled person will also understand that any use of "or" throughout the statements of invention or description herein encompasses use of "or", "and/or", and "and". For example, the term "or" within the discourse is construed to encompass both "and" and "and/or" owing to its inherent inclusivity. Within linguistic reasoning, "or" denotes an inclusive disjunction, allowing for the consideration of scenarios wherein either one condition holds true, the other condition holds true, or both conditions hold true concurrently. This interpretation inherently incorporates the conjunction "and", permitting the acknowledgment of scenarios wherein multiple conditions coexist. Additionally, the term "and/or" explicitly acknowledges the possibility of either condition being singularly true or both conditions being true simultaneously, thus aligning with the broader meaning of "or" within the context of this disclosure. Consequently, "or" functions as a flexible connector within the statements of invention, accommodating both exclusive and inclusive interpretations to suit the nuanced requirements of embodiments described herein.

20

PATENT CLAIMS

1. A method for media content identification and content information retrieval at a user device, the method comprising:

monitoring, using a microphone of the user device, a portion of an audio track emitting from a speaker, wherein the audio track is associated with a media content;
detecting whether there is an audio tag embedded within the portion of the audio track, wherein the audio tag is embedded at frequencies detectable by the microphone and less detectable to a user of the user device;

if an audio tag is detected:

extracting identification data from the audio tag, wherein the identification data is associated with a known media content at a known timepoint;

if an audio tag is not detected:

receiving identification data based on a best match for an audio fingerprint from a plurality of audio fingerprints, wherein the audio fingerprint comprises one or more audio features extracted from the portion of the audio track;

identifying the media content based on the known media content of the identification data; and

retrieving content information associated with the identification data.

2. The method of claim 1, wherein retrieving content information comprises:

transmitting, to a server device, an information request comprising the identification data; and

obtaining, from the server device, information associated with one or more subjects relevant to the known media content at the known timepoint, wherein each subject of the one or more subjects is linked to the known timepoint.

3. The method of claim 2, wherein the information request is transmitted based on a user interaction with the user device.

4. The method of any preceding claim, wherein the speaker is associated with a media device independent of the user device.

5. The method of any preceding claim, further comprising:

identifying a timepoint of the media content based on the known timepoint of the identification data; and

displaying content information on the user device based on the timepoint, such that displayed content information is relevant to the media content at the known timepoint.

6. The method of any preceding claim, wherein the microphone is switched on or off at periodic intervals while monitoring the portion of the audio track.

7. The method of any preceding claim, wherein the audio tag is embedded using frequency-shift keying modulation at defined intervals.

8. The method of claim 7, wherein the defined intervals are adjusted after the media content is identified.
9. The method of any preceding claim, wherein the known timepoint is a predetermined time interval.
10. The method of any preceding claim, wherein the plurality of audio fingerprints comprises:
 - audio fingerprints associated with a plurality of portions of an audio track at a plurality of time points; or
 - audio fingerprints associated with one or more portions of a plurality of audio tracks.
11. The method of any preceding claim, wherein receiving identification data based on the best match comprises:
 - transmitting the audio fingerprint or the portion of the audio track to a server device, wherein the audio fingerprint is generated by extracting the one or more audio features from the portion of the audio track; and
 - obtaining, from the server device, identification data associated with the best match, wherein the server device is configured to compare the audio fingerprint against the plurality of audio fingerprints stored in a database accessible to the server device, thereby determining the best match.
12. The method of any preceding claim, wherein detecting whether there is an audio tag embedded within the portion of the audio track comprises application of one or more correction algorithms to compensate for noise, interference, or other audio degradation factors.
13. The method of any preceding claim, wherein detecting whether there is an audio tag is limited to a predefined time frame.
14. A computer-readable medium storing instructions for performing the method of any of claims 1-13.
15. A user device comprising:
 - a microphone;
 - a user interface; and
 - a user application installed on the user device providing instructions that, when executed, cause the user device to perform the method of any of claims 1-13.