(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
07 March 2019 (07.03.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/046790 A1

(51) International Patent Classification:
H04W 4/33 (2018.01) *G06T 11/60* (2006.01)
G06K 9/00 (2006.01) *H04L 12/58* (2006.01)
G06K 9/62 (2006.01) *H04W 88/02* (2009.01)

(21) International Application Number:
PCT/US2018/049213

(22) International Filing Date:
31 August 2018 (31.08.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/692,990 31 August 2017 (31.08.2017) US
15/967,201 30 April 2018 (30.04.2018) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 15/967,201 (CON)
Filed on 30 April 2018 (30.04.2018)

(71) Applicant: **SNAP INC.** [US/US]; 2772 Donald Douglas Loop North, Santa Monica, California 90405 (US).

(72) Inventors: **CHARLTON, Ebony James**; 2772 Donald Douglas Loop North, Santa Monica, California 90405 (US). **HANUMANTE, Sumant Milind**; 2772 Donald Douglas Loop North, Santa Monica, California 90405 (US). **REN, Zhou**; 2772 Donald Douglas Loop North, Santa Monica, California 90405 (US). **SAGAR, Dhritiman**; 2772 Donald Douglas Loop North, Santa Monica, California 90405 (US).

(74) Agent: **PERDOK, Monique M.** et al.; SCHWEGMAN LUNDBERG & WOESSNER P.A., P.O. Box 2938, Minneapolis, Minnesota 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,

(54) Title: DEVICE LOCATION BASED ON MACHINE LEARNING CLASSIFICATIONS

(57) Abstract: A venue system of a client device can submit a location request to a server, which returns multiple venues that are near the client device. The client device can use one or more machine learning schemes (e.g., convolutional neural networks) to determine that the client device is located in one of specific venues of the possible venues. The venue system can further select imagery for presentation based on the venue selection. The presentation may be published as ephemeral message on a network platform.

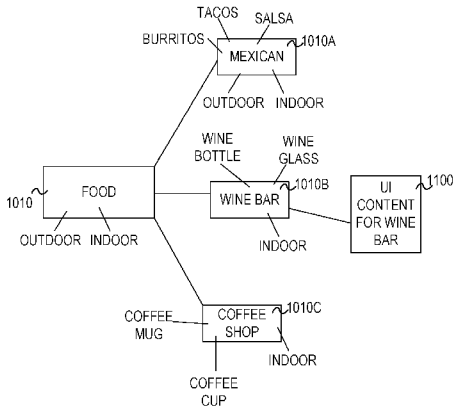


FIG. 11

[Continued on next page]

WO 2019/046790 A1

MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

DEVICE LOCATION BASED ON MACHINE LEARNING CLASSIFICATIONS

PRIORITY

[0001] This application claims the benefit of priority to U.S. Patent Application Serial No. 15/967,201, filed on April 30, 2018, which is a continuation of and claims the benefit of priority to U.S. Patent Application Serial No. 15/692,990, filed August 31, 2017, the benefit of priority of each of which is claimed hereby, and each of which is incorporated by reference herein in its entirety.

TECHNICAL FIELD

[0002] Embodiments of the present disclosure relate generally to determining device locations and, more particularly, but not by way of limitation, to precise computer device location determinations based on machine learning classifications.

BACKGROUND

[0003] A computer (e.g., a smartphone) can approximate its location using geolocation services. For example, the computer can use global positioning system (GPS) data generated by an onboard GPS sensor to determine its location. However, often such information cannot pinpoint in which venue (e.g., café, restaurant, gas station) the computer is currently located if there are several venues near the computer.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number

refer to the figure ("FIG.") number in which that element or act is first introduced.

[0005] FIG. 1 is a block diagram showing an example messaging system for exchanging data (e.g., messages and associated content) over a network.

[0006] FIG. 2 is block diagram illustrating further details regarding a messaging system having an integrated virtual object machine learning system, according to example embodiments.

[0007] FIG. 3 is a schematic diagram illustrating data which may be stored in a database of a messaging server system, according to certain example embodiments.

[0008] FIG. 4 is a schematic diagram illustrating a structure of a message, according to some embodiments, generated by a messaging client application for communication.

[0009] FIG. 5 is a schematic diagram illustrating an example access-limiting process, in terms of which access to content (e.g., an ephemeral message, and associated multimedia payload of data) or a content collection (e.g., an ephemeral message story) may be time-limited (e.g., made ephemeral).

[0010] FIG. 6A shows example components of a client venue system, according to some example embodiments.

[0011] FIG. 6B shows example components of a server venue system, according to some example embodiments.

[0012] FIG. 7 shows a flow diagram of a method for implementing improved venue selection using machine learning classifications, according to some example embodiments.

[0013] FIG. 8 shows a flow diagram of a method of generating classifications for images, according to some example embodiments.

[0014] FIG. 9 shows a flow diagram of a method for selecting one or more venues from a venue dataset, according to some example embodiments.

[0015] FIG. 10 shows a data structure that can be used to filter or otherwise select venues and display content, according to some example embodiments.

[0016] FIG. 11 shows an example of metadata tags with a portion of the data structure, according to some example embodiments.

[0017] FIG. 12 shows an example presentation of an image of a wine bar with overlay content, according to some example embodiments.

[0018] FIG. 13 shows an example presentation of a casino with overlay content, according to some example embodiments.

[0019] FIG. 14 shows an example presentation of a pool area with overlay content, according to some example embodiments.

[0020] FIG. 15 is a block diagram illustrating a representative software architecture, which may be used in conjunction with various hardware architectures herein described.

[0021] FIG. 16 is a block diagram illustrating components of a machine, according to some example embodiments, able to read instructions from a machine-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein.

DETAILED DESCRIPTION

[0022] The description that follows includes systems, methods, techniques, instruction sequences, and computing machine program products that embody illustrative embodiments of the disclosure. In the following description, for the purposes of explanation, numerous specific details are set forth in order to

provide an understanding of various embodiments of the inventive subject matter. It will be evident, however, to those skilled in the art, that embodiments of the inventive subject matter may be practiced without these specific details. In general, well-known instruction instances, protocols, structures, and techniques are not necessarily shown in detail.

[0023] A client device, such as a smartphone, can approximate its location using geolocation services. For example, a smartphone can use an onboard GPS sensor to determine the smartphones latitude and longitude. However, GPS accuracy can be limited by environmental factors, such as obstructing buildings and other objects that interfere with GPS signals. For example, due to physical obstructions (e.g., physical walls) a client device's GPS sensor may show the device at one restaurant when in fact the client device is located in another restaurant hundreds of feet away. Further, client device GPS accuracy can also be limited by protocol. For instance, while military grade GPS can be highly accurate, consumer-grade GPS often has distance resolution restrictions so the consumer-grade GPS devices are not used for malicious purposes. For these reasons, client device generated geolocation data may not accurately describe in which venue (e.g., café, restaurant, gas station) the client device is currently located.

[0024] To attempt to improve location determinations, the client device can send the GPS data to a server for further analysis. However, if the client device GPS signal is poor or if there are many venues nearby the data sent to the server may be inaccurate, in which case subsequent predictions generated by the server will also be inaccurate.

[0025] To this end, a venue system executing on a client device can leverage visual cues of the environment surrounding the client device to select a given venue from a set of possible venues. The

visual cues may be identified using a machine learning scheme, such as one or more neural networks that have been trained for image processing (e.g., convolutional neural networks). Initially, the client device may collect its geolocation data and other information potentially helpful for location identification, e.g., a list of Internet Protocol (IP) networks that are visible to the client device. The client device can send the collected information to a location server for further analysis.

[0026] The server receives the request, accesses a venue database and returns the a number of venues that are nearest to the client device (e.g., nearest five venues). The venue database accessed by the server may contain location data (e.g., latitude/longitude data, address data, visible IP network data) for venues anywhere in the world. The server can use the geolocation data and other data (e.g., visible IP networks) to determine which venues in the database are closest to the client device.

[0027] According to some example embodiments, in response to receiving the set of venues from the server, the client device then attempts to filter out venues from the set of venues using visual cues from images of the surrounding environment. In particular, for example, the venue system implements one or more trained neural networks to analyze image data generated by the client device. For example, the venue system may use the trained neural networks to analyze images (e.g., an image, video, live video feed) captured by the user using his/her smartphone. This approach leverages the fact that the server has access to a potentially very large database of venues that is not storable on the client device and further leverages the client device's access to visual cues of the surrounding environment.

[0028] In some example embodiments, the venue system on the client device uses up to three machine learning schemes to

determine visual information for venue selection. In those example embodiments, a first learning scheme is configured to determine whether the client device is in an outside environment or inside environment, a second machine learning scheme is configured to determine the type of venue directly, and a third machine learning scheme is configured to recognize objects (e.g., coffee mugs, slices of pizza, etc.) in a given venue. Further details of the engines are discussed below with reference to FIG. 6A.

[0029] An example can help illustrate how the machine learning scheme and accompanying logic are implemented, according to some example embodiments. Assume the client device is in a coffee shop and the user uses a smartphone to image the coffee shop, e.g., by taking a picture or video feed of the coffee shop. In response to imaging the coffee shop, the venue system automatically sends a location request to the server and receives four nearby venues as potential locations of the client device. Assume the four nearby venues include: an outdoor rooftop restaurant, headquarters of a social media technology company, an Italian restaurant, and a coffee shop.

[0030] The venue system then applies the machine learning schemes to the image to classify attributes about the image. In particular, for example, the first machine learning scheme determines that the image is of an inside environment, the second machine learning scheme determines that the image is most likely an Italian restaurant or a coffee shop, and the third machine learning scheme identifies a coffee mug in the image. Based on these determinations, the venue system excludes the outdoor restaurant because it is outside and the imaged environment has been determined to be inside by the first machine learning scheme. The second machine learning scheme determined that there is an equal likelihood that the image is of an Italian restaurant or of a

coffee shop, and a low likelihood that the image is of an office setting (e.g., a technology company's headquarters), accordingly the headquarters is filtered or otherwise not selected. Finally, based on the third machine learning system identifying a coffee cup in the image, the venue system selects the coffee shop as the most likely venue for the client device's current location.

[0031] Each venue may be categorized in a data structure that includes categories and sub-categories, according to some example embodiments. The categories contain venue types at a higher level of granularity and the sub-categories refine a corresponding venue type. For example, a food category may have sub-categories including Mexican restaurant, fast food restaurant, Italian restaurant, coffee shop, and so on. Other categories and sub-categories may similarly be configured. Each specific venue can be generalized to a broader level which can more readily be tagged. For example, "Tony's Family Italian Restaurant – World's Best!" can be abstracted to the Italian restaurant venue type, which can be more readily be assigned tags and stored on the limited memory of a client device (e.g., a smartphone).

[0032] Each category and subcategory may have metadata tags that describe the category or subcategory. The metadata tags may include environmental conditions (e.g., "outside," "inside") as well as objects associated with a given type of venue (e.g., an Italian food restaurant sub-category may have tags including "pizza," "red and white checkered pattern," and "spaghetti"; while a coffee shop sub-category may have tags including "coffee mug," "coffee bean," and "Starbucks logo".)

[0033] The venue system can use the classifications generated by the machine learning schemes to select matching venues. Referring to the above example, the venue system filtered out venues having "outdoor" tags because the image was determined to

be of an inside environment. Further, the second machine learning scheme a low likelihood that that the image is of an office, and returned equal numerical likelihoods for Italian restaurant and coffee shop; accordingly, the venue system can filter out or otherwise not select the technology company headquarters venue. The third machine learning scheme successfully identified a coffee mug in the image, and coffee mug is a metadata tag of the coffee shop category, thus the coffee shop is selected as the current venue of the client device. Other examples are discussed below with reference to the figures.

[0034] Once the venue is selected, the venue system or other applications on the client device can use the selected venue to create location based user interface (UI) content. Continuing the example, once the coffee shop is determined, a cartoon avatar (e.g., bitmoji) of the user operating the client device can be overlaid on the image of the coffee shop. The image with the overlaid content can the be posted to a social media network, e.g., as an ephemeral message, discussed in further detail below. Other examples of UI content include banners, captions (e.g., a caption outside the border of the image denoting where the image was taken), or augmented reality content, and so on. The UI content can be categorized using the same data structure of categories and sub-categories used to structure the venue types. In this way, when a venue of a given category (or sub-category) is selected, the UI content for overlay can be readily retrieved.

[0035] FIG. 1 is a block diagram showing an example messaging system 100 for exchanging data (e.g., messages and associated content) over a network. The messaging system 100 includes multiple client devices 102, each of which hosts a number of applications including a messaging client application 104. Each messaging client application 104 is communicatively coupled to

other instances of the messaging client application 104 and a messaging server system 108 via a network 106 (e.g., the Internet).

[0036] Accordingly, each messaging client application 104 is able to communicate and exchange data with another messaging client application 104 and with the messaging server system 108 via the network 106. The data exchanged between messaging client devices 102, and between a messaging client application 104 and the messaging server system 108, includes functions (e.g., commands to invoke functions) as well as payload data (e.g., text, audio, video, or other multimedia data).

[0037] The messaging server system 108 provides server-side functionality via the network 106 to a particular messaging client application 104. While certain functions of the messaging system 100 are described herein as being performed by either a messaging client application 104 or by the messaging server system 108, it will be appreciated that the location of certain functionality within either the messaging client application 104 or the messaging server system 108 is a design choice. For example, it may be technically preferable to initially deploy certain technology and functionality within the messaging server system 108, and to later migrate this technology and functionality to the messaging client application 104 where a client device 102 has a sufficient processing capacity.

[0038] The messaging server system 108 supports various services and operations that are provided to the messaging client application 104. Such operations include transmitting data to, receiving data from, and processing data generated by the messaging client application 104. This data may include message content, client device information, geolocation information, media annotation and overlays, message content persistence conditions, social network information, and live event information, as

examples. Data exchanges within the messaging system 100 are invoked and controlled through functions available via user interfaces (UIs) of the messaging client application 104.

[0039] Turning now specifically to the messaging server system 108, an application programming interface (API) server 110 is coupled to, and provides a programmatic interface to, an application server 112. The application server 112 is communicatively coupled to a database server 118, which facilitates access to a database 120 in which is stored data associated with messages processed by the application server 112.

[0040] The API server 110 receives and transmits message data (e.g., commands and message payloads) between the client devices 102 and the application server 112. Specifically, the API server 110 provides a set of interfaces (e.g., routines and protocols) that can be called or queried by the messaging client application 104 in order to invoke functionality of the application server 112. The API server 110 exposes various functions supported by the application server 112, including account registration; login functionality; the sending of messages, via the application server 112, from a particular messaging client application 104 to another messaging client application 104; the sending of media files (e.g., images or video) from a messaging client application 104 to a messaging server application 114 for possible access by another messaging client application 104; the setting of a collection of media data (e.g., a story); the retrieval of such collections; the retrieval of a list of friends of a user of a client device 102; the retrieval of messages and content; the adding and deletion of friends to and from a social graph; the location of friends within the social graph; and opening application events (e.g., relating to the messaging client application 104).

[0041] The application server 112 hosts a number of applications and subsystems, including the messaging server application 114, an image processing system 116, a social network system 122, and a server venue system 150. The messaging server application 114 implements a number of message-processing technologies and functions, particularly related to the aggregation and other processing of content (e.g., textual and multimedia content) included in messages received from multiple instances of the messaging client application 104. As will be described in further detail, the text and media content from multiple sources may be aggregated into collections of content (e.g., called stories or galleries). These collections are then made available, by the messaging server application 114, to the messaging client application 104. Other processor- and memory-intensive processing of data may also be performed server-side by the messaging server application 114, in view of the hardware requirements for such processing.

[0042] The application server 112 also includes the image processing system 116, which is dedicated to performing various image processing operations, typically with respect to images or video received within the payload of a message at the messaging server application 114.

[0043] The social network system 122 supports various social networking functions and services, and makes these functions and services available to the messaging server application 114. To this end, the social network system 122 maintains and accesses an entity graph (e.g., entity graph 304 in FIG. 3) within the database 120. Examples of functions and services supported by the social network system 122 include the identification of other users of the messaging system 100 with whom a particular user has relationships or whom the particular user is "following," and also

the identification of other entities and interests of a particular user.

[0044] The server venue system 150 is configured to receive, from a client device 102, a request for venues near the location of the client device 102. Further details of the server venue system 150 are discussed below with reference to FIG. 6B.

[0045] The application server 112 is communicatively coupled to a database server 118, which facilitates access to a database 120 in which is stored data associated with messages processed by the messaging server application 114.

[0046] FIG. 2 is block diagram illustrating further details regarding the messaging system 100, according to example embodiments. Specifically, the messaging system 100 is shown to comprise the messaging client application 104 and the application server 112, which in turn embody a number of subsystems, namely an ephemeral timer system 202, a collection management system 204, an annotation system 206, and a client venue system 210.

[0047] The ephemeral timer system 202 is responsible for enforcing the temporary access to content permitted by the messaging client application 104 and the messaging server application 114. To this end, the ephemeral timer system 202 incorporates a number of timers that, based on duration and display parameters associated with a message or collection of messages (e.g., a story), selectively display and enable access to messages and associated content via the messaging client application 104. Further details regarding the operation of the ephemeral timer system 202 are provided below.

[0048] The collection management system 204 is responsible for managing collections of media (e.g., collections of text, image, video, and audio data). In some examples, a collection of content (e.g., messages, including images, video, text, and audio) may be

organized into an "event gallery" or an "event story." Such a collection may be made available for a specified time period, such as the duration of an event to which the content relates. For example, content relating to a music concert may be made available as a "story" for the duration of that music concert. The collection management system 204 may also be responsible for publishing an icon that provides notification of the existence of a particular collection to the user interface of the messaging client application 104.

[0049] The collection management system 204 furthermore includes a curation interface 208 that allows a collection manager to manage and curate a particular collection of content. For example, the curation interface 208 enables an event organizer to curate a collection of content relating to a specific event (e.g., delete inappropriate content or redundant messages). Additionally, the collection management system 204 employs machine vision (or image recognition technology) and content rules to automatically curate a content collection. In certain embodiments, compensation may be paid to a user for inclusion of user-generated content into a collection. In such cases, the curation interface 208 operates to automatically make payments to such users for the use of their content.

[0050] The annotation system 206 provides various functions that enable a user to annotate or otherwise modify or edit media content associated with a message. For example, the annotation system 206 provides functions related to the generation and publishing of media overlays for messages processed by the messaging system 100. The annotation system 206 operatively supplies a media overlay (e.g., a geofilter or filter) to the messaging client application 104 based on a geolocation of the client device 102. In another example, the annotation system 206

operatively supplies a media overlay to the messaging client application 104 based on other information, such as social network information of the user of the client device 102 or a venue selected by the client venue system 210. A media overlay may include audio and visual content and visual effects. Examples of audio and visual content include pictures, text, logos, animations, and sound effects. An example of a visual effect includes color overlaying. The audio and visual content or the visual effects can be applied to a media content item (e.g., a photo) at the client device 102. For example, the media overlay includes text that can be overlaid on top of a photograph generated by the client device 102. In another example, the media overlay includes an identification of a location (e.g., Venice Beach), a name of a live event, or a name of a merchant (e.g., Beach Coffee House). In another example, the annotation system 206 uses the geolocation of the client device 102 to identify a media overlay that includes the name of a merchant at the geolocation of the client device 102. The media overlay may include other indicia associated with the merchant. The media overlays may be stored in the database 120 and accessed through the database server 118.

[0051] In one example embodiment, the annotation system 206 provides a user-based publication platform that enables users to select a geolocation on a map and upload content associated with the selected geolocation. The user may also specify circumstances under which particular content should be offered to other users. The annotation system 206 generates a media overlay that includes the uploaded content and associates the uploaded content with the selected geolocation.

[0052] In another example embodiment, the annotation system 206 provides a merchant-based publication platform that enables merchants to select a particular media overlay associated with a

geolocation via a bidding process. For example, the annotation system 206 associates the media overlay of a highest-bidding merchant with a corresponding geolocation for a predefined amount of time.

[0053] FIG. 3 is a schematic diagram illustrating data 300 which may be stored in the database 120 of the messaging server system 108, according to certain example embodiments. While the content of the database 120 is shown to comprise a number of tables, it will be appreciated that the data could be stored in other types of data structures (e.g., as an object-oriented database).

[0054] The database 120 includes message data stored within a message table 314. An entity table 302 stores entity data, including an entity graph 304. Entities for which records are maintained within the entity table 302 may include individuals, corporate entities, organizations, objects, places, events, and so forth. Regardless of type, any entity regarding which the messaging server system 108 stores data may be a recognized entity. Each entity is provided with a unique identifier, as well as an entity type identifier (not shown).

[0055] The entity graph 304 furthermore stores information regarding relationships and associations between or among entities. Such relationships may be social, professional (e.g., work at a common corporation or organization), interest-based, or activity-based, for example.

[0056] The database 120 also stores annotation data, in the example form of filters, in an annotation table 312. Filters for which data is stored within the annotation table 312 are associated with and applied to videos (for which data is stored in a video table 310) and/or images (for which data is stored in an image table 308). Filters, in one example, are overlays that are displayed as overlaid on an image or video during presentation to a recipient

user. Filters may be of various types, including user-selected filters from a gallery of filters presented to a sending user by the messaging client application 104 when the sending user is composing a message. Other types of filters include geolocation filters (also known as geo-filters), which may be presented to a sending user based on geographic location. For example, geolocation filters specific to a neighborhood or special location may be presented within a user interface by the messaging client application 104, based on geolocation information determined by a Global Positioning System (GPS) unit of the client device 102. Another type of filter is a data filter, which may be selectively presented to a sending user by the messaging client application 104, based on other inputs or information gathered by the client device 102 during the message creation process. Examples of data filters include a current temperature at a specific location, a current speed at which a sending user is traveling, a battery life for a client device 102, or the current time.

[0057] Other annotation data that may be stored within the image table 308 is so-called "lens" data. A "lens" may be a real-time special effect and sound that may be added to an image or a video.

[0058] As mentioned above, the video table 310 stores video data which, in one embodiment, is associated with messages for which records are maintained within the message table 314. Similarly, the image table 308 stores image data associated with messages for which message data is stored in the message table 314. The entity table 302 may associate various annotations from the annotation table 312 with various images and videos stored in the image table 308 and the video table 310.

[0059] A story table 306 stores data regarding collections of messages and associated image, video, or audio data, which are compiled into a collection (e.g., a story or a gallery). The creation of

a particular collection may be initiated by a particular user (e.g., each user for whom a record is maintained in the entity table 302). A user may create a "personal story" in the form of a collection of content that has been created and sent/broadcast by that user. To this end, the user interface of the messaging client application 104 may include an icon that is user-selectable to enable a sending user to add specific content to his or her personal story.

[0060] A collection may also constitute a "live story," which is a collection of content from multiple users that is created manually, automatically, or using a combination of manual and automatic techniques. For example, a "live story" may constitute a curated stream of user-submitted content from various locations and events. Users whose client devices 102 have location services enabled and are at a common location or event at a particular time may, for example, be presented with an option, via a user interface of the messaging client application 104, to contribute content to a particular live story. The live story may be identified to the user by the messaging client application 104 based on his or her location. The end result is a "live story" told from a community perspective.

[0061] A further type of content collection is known as a "location story," which enables a user whose client device 102 is located within a specific geographic location (e.g., on a college or university campus) to contribute to a particular collection. In some embodiments, a contribution to a location story may require a second degree of authentication to verify that the end user belongs to a specific organization or other entity (e.g., is a student on the university campus).

[0062] FIG. 4 is a schematic diagram illustrating a structure of a message 400, according to some embodiments, generated by a messaging client application 104 for communication to a further messaging client application 104 or the messaging server

application 114. The content of a particular message 400 is used to populate the message table 314 stored within the database 120, accessible by the messaging server application 114. Similarly, the content of a message 400 is stored in memory as "in-transit" or "in-flight" data of the client device 102 or the application server 112. The message 400 is shown to include the following components:

- A message identifier 402: a unique identifier that identifies the message 400.
- A message text payload 404: text, to be generated by a user via a user interface of the client device 102 and that is included in the message 400.
- A message image payload 406: image data captured by a camera component of a client device 102 or retrieved from memory of a client device 102, and that is included in the message 400.
- A message video payload 408: video data captured by a camera component or retrieved from a memory component of the client device 102, and that is included in the message 400.
- A message audio payload 410: audio data captured by a microphone or retrieved from the memory component of the client device 102, and that is included in the message 400.
- Message annotations 412: annotation data (e.g., filters, stickers, or other enhancements) that represents annotations to be applied to the message image payload 406, message video payload 408, or message audio payload 410 of the message 400.
- A message duration parameter 414: a parameter value indicating, in seconds, the amount of time for which content of the message 400 (e.g., the message image payload 406, message video payload 408, and message audio payload 410) is to be presented or made accessible to a user via the messaging client application 104.

- A message geolocation parameter 416: geolocation data (e.g., latitudinal and longitudinal coordinates) associated with the content payload of the message 400. Multiple message geolocation parameter 416 values may be included in the payload, with each of these parameter values being associated with respective content items included in the content (e.g., a specific image in the message image payload 406, or a specific video in the message video payload 408).
- A message story identifier 418: identifies values identifying one or more content collections (e.g., "stories") with which a particular content item in the message image payload 406 of the message 400 is associated. For example, multiple images within the message image payload 406 may each be associated with multiple content collections using identifier values.
- A message tag 420: one or more tags, each of which is indicative of the subject matter of content included in the message payload. For example, where a particular image included in the message image payload 406 depicts an animal (e.g., a lion), a tag value may be included within the message tag 420 that is indicative of the relevant animal. Tag values may be generated manually, based on user input, or may be automatically generated using, for example, image recognition.
- A message sender identifier 422: an identifier (e.g., a messaging system identifier, email address, or device identifier) indicative of a user of the client device 102 on which the message 400 was generated and from which the message 400 was sent.
- A message receiver identifier 424: an identifier (e.g., a messaging system identifier, email address, or device identifier) indicative of a user of the client device 102 to which the message 400 is addressed.

[0063] The contents (e.g., values) of the various components of the message 400 may be pointers to locations in tables within which content data values are stored. For example, an image value in the message image payload 406 may be a pointer to (or address of) a location within the image table 308. Similarly, values within the message video payload 408 may point to data stored within the video table 310, values stored within the message annotations 412 may point to data stored in the annotation table 312, values stored within the message story identifier 418 may point to data stored in the story table 306, and values stored within the message sender identifier 422 and the message receiver identifier 424 may point to user records stored within the entity table 302.

[0064] FIG. 5 is a schematic diagram illustrating an access-limiting process 500, in terms of which access to content (e.g., an ephemeral message 502, and associated multimedia payload of data) or a content collection (e.g., an ephemeral message story 504) may be time-limited (e.g., made ephemeral).

[0065] An ephemeral message 502 is shown to be associated with a message duration parameter 506, the value of which determines an amount of time that the ephemeral message 502 will be displayed to a receiving user of the ephemeral message 502 by the messaging client application 104. In one embodiment, where the messaging client application 104 is a application client, an ephemeral message 502 is viewable by a receiving user for up to a maximum of 10 seconds, depending on the amount of time that the sending user specifies using the message duration parameter 506.

[0066] The message duration parameter 506 and the message receiver identifier 424 are shown to be inputs to a message timer 512, which is responsible for determining the amount of time that the ephemeral message 502 is shown to a particular receiving user identified by the message receiver identifier 424. In particular, the

ephemeral message 502 will only be shown to the relevant receiving user for a time period determined by the value of the message duration parameter 506. The message timer 512 is shown to provide output to a more generalized ephemeral timer system 202, which is responsible for the overall timing of display of content (e.g., an ephemeral message 502) to a receiving user.

[0067] The ephemeral message 502 is shown in FIG. 5 to be included within an ephemeral message story 504 (e.g., a personal story, or an event story). The ephemeral message story 504 has an associated story duration parameter 508, a value of which determines a time duration for which the ephemeral message story 504 is presented and accessible to users of the messaging system 100. The story duration parameter 508, for example, may be the duration of a music concert, where the ephemeral message story 504 is a collection of content pertaining to that concert. Alternatively, a user (either the owning user or a curator user) may specify the value for the story duration parameter 508 when performing the setup and creation of the ephemeral message story 504.

[0068] Additionally, each ephemeral message 502 within the ephemeral message story 504 has an associated story participation parameter 510, a value of which determines the duration of time for which the ephemeral message 502 will be accessible within the context of the ephemeral message story 504. Accordingly, a particular ephemeral message 502 may "expire" and become inaccessible within the context of the ephemeral message story 504, prior to the ephemeral message story 504 itself expiring in terms of the story duration parameter 508. The story duration parameter 508, story participation parameter 510, and message receiver identifier 424 each provide input to a story timer 514, which operationally determines whether a particular ephemeral

message 502 of the ephemeral message story 504 will be displayed to a particular receiving user and, if so, for how long. Note that the ephemeral message story 504 is also aware of the identity of the particular receiving user as a result of the message receiver identifier 424.

[0069] Accordingly, the story timer 514 operationally controls the overall lifespan of an associated ephemeral message story 504, as well as an individual ephemeral message 502 included in the ephemeral message story 504. In one embodiment, each and every ephemeral message 502 within the ephemeral message story 504 remains viewable and accessible for a time period specified by the story duration parameter 508. In a further embodiment, a certain ephemeral message 502 may expire, within the context of the ephemeral message story 504, based on a story participation parameter 510. Note that a message duration parameter 506 may still determine the duration of time for which a particular ephemeral message 502 is displayed to a receiving user, even within the context of the ephemeral message story 504. Accordingly, the message duration parameter 506 determines the duration of time that a particular ephemeral message 502 is displayed to a receiving user, regardless of whether the receiving user is viewing that ephemeral message 502 inside or outside the context of an ephemeral message story 504.

[0070] The ephemeral timer system 202 may furthermore operationally remove a particular ephemeral message 502 from the ephemeral message story 504 based on a determination that it has exceeded an associated story participation parameter 510. For example, when a sending user has established a story participation parameter 510 of 24 hours from posting, the ephemeral timer system 202 will remove the relevant ephemeral message 502 from the ephemeral message story 504 after the specified 24 hours. The

ephemeral timer system 202 also operates to remove an ephemeral message story 504 either when the story participation parameter 510 for each and every ephemeral message 502 within the ephemeral message story 504 has expired, or when the ephemeral message story 504 itself has expired in terms of the story duration parameter 508.

[0071] In certain use cases, a creator of a particular ephemeral message story 504 may specify an indefinite story duration parameter 508. In this case, the expiration of the story participation parameter 510 for the last remaining ephemeral message 502 within the ephemeral message story 504 will determine when the ephemeral message story 504 itself expires. In this case, a new ephemeral message 502, added to the ephemeral message story 504, with a new story participation parameter 510, effectively extends the life of an ephemeral message story 504 to equal the value of the story participation parameter 510.

[0072] In response to the ephemeral timer system 202 determining that an ephemeral message story 504 has expired (e.g., is no longer accessible), the ephemeral timer system 202 communicates with the messaging system 100 (e.g., specifically, the messaging client application 104) to cause an indicium (e.g., an icon) associated with the relevant ephemeral message story 504 to no longer be displayed within a user interface of the messaging client application 104. Similarly, when the ephemeral timer system 202 determines that the message duration parameter 506 for a particular ephemeral message 502 has expired, the ephemeral timer system 202 causes the messaging client application 104 to no longer display an indicium (e.g., an icon or textual identification) associated with the ephemeral message 502.

[0073] FIG. 6A shows example internal functional components of a client venue system 210, according to some example embodiments.

As illustrated, the client venue system 210 comprises a location engine 600, an image engine 605, an environment classification engine 610, a venue classification engine 615, an object classification engine 620, a selection engine 625, and a display engine 630. The location engine 600 is configured to transmit a location request to a location server, according to some example embodiments. For example, the request may contain latitude and longitude information generated from an onboard GPS sensor of a client device, e.g., client device 102. The location server (see FIG. 6B) can use the latitude and longitude information to determine which venues are nearest to the client device and send the nearest venues (e.g., up to the nearest five venues) to the client device. In some example embodiments, the request further includes IP networks that are visible to the client device. In those example embodiments, the location server can use the name or type of visible IP networks to further ascertain which venues are nearest to the client device.

[0074] The image engine 605 manages identifying one or more images using an image sensor of the client device 102. In some example embodiments, the image engine 605 identifies images already generated by the messaging client application 104 (and stored in image table 308) and imports them into the client venue system 210 for further processing. In some example embodiments, the image engine 605 interfaces with an image sensor of the client device 102 to generate one or more images. Further, in any of the example embodiments, the images identified by the image engine 605 may be part of a live video feed which is being dynamically displayed on a screen of the client device 102 as the video is recorded.

[0075] The environment classification engine 610 is configured to receive the images identified by the image engine 605 and classify

each image as being an image of an outdoor environment or an image of an indoor environment. The classification generated by the environment classification engine 610 may be stored as metadata of each classified image (e.g., as metadata of the image file, or separate data that references an image file). In some example embodiments, the environment classification engine 610 is implemented as a machine learning scheme, such as a convolutional neural network. The convolutional neural network in environment classification engine 610 can be trained on a training set of images of outdoor and indoor environments. Examples of outdoor environments imaged in the training set include parks, streets, airplane interiors, amphitheaters, and so on. Examples of indoor environments imaged in the training set includes bowling alleys, bedrooms, kitchens, classrooms, ballrooms, offices, theaters, and so on. In some example embodiments, the classifications output by the environment classification engine 610 are output as numerical likelihood that a given image is of a particular environment, as is appreciated by those of ordinary skill in the art (e.g., an image of a park may yield the following classification likelihoods: outside = 0.93, inside = 0.20, where the decimals corresponding percentages).

[0076] The venue classification engine 615 is configured to receive the images identified by the image engine 605 and classify each image as a type of venue. The classification generated by the venue classification engine 615 may be stored as metadata of each classified image file or as separate data that references the image file. In some example embodiments, the venue classification engine 615 is implemented as a machine learning scheme, such as a convolutional neural network. The convolutional neural network in the venue classification engine 615 can be trained on the training set of images of different types of venues or places. In contrast

with the environment classification engine 610 (which outputs likelihoods of indoor and outdoor environments), the venue classification engine 615 is configured to output likelihoods that a given image is an image of known specific places (e.g., bar, Italian restaurant, park, beach). The convolutional neural network in the venue classification engine 615 can be trained on a training set of images including images of the different places, e.g., an image of a bar, an image of a beach, an image of a casino, an image of a mountain range. In some example embodiments, the classifications output by the venue classification engine 615 are output as a numerical likelihood that a given image is of a particular venue, as is appreciated by those of ordinary skill in the art (e.g., an image of a beach may yield the following classification likelihoods: beach = 0.83, yard = 0.75, bar = 0.34, Italian restaurant = 0.30).

[0077] The object classification engine 620 is configured to classify physical objects depicted in the images. The classification generated by the object classification engine 620 may be stored as metadata of each classified image file or as separate data that references a corresponding image file. In some example embodiments, the object classification engine 620 is implemented as a machine learning scheme, such as a convolutional neural network. The convolutional neural network in the object classification engine 620 can be trained on the training set of images of different types of physical objects, such as coffee mugs, wine glasses, pizza, Christmas trees, swimming pools, vehicles, animals, and so on. In some example embodiments, the classifications output by the object classification engine 620 are output as numerical likelihood that a given image feature in an image is of a particular object, as is appreciated by those of ordinary skill in the art (e.g., a wine glass image feature within a bounded region of interest (ROI) in a given image may have the

following classification likelihoods: wine glass = 0.93, frog = 0.05, basketball = 0.30, coffee mug = 0.60).

[0078] The selection engine 625 manages selecting a venue from a venue data set (e.g. a venue data set returned from an application server) using the classification data generated by the other engines, according to some example embodiments. Further, the selection engine 625 may further be configured to select one or more UI display elements based on the type of venue selected. For example, if the selection engine 625 determines that the client device 102 is in a bar (e.g., a bar sub-category), the selection engine 625 automatically selects a cartoon avatar and optional lens data for overlay on the one or more images.

[0079] The display engine 630 is configured to generate a presentation of the one or more images (e.g. an image, a live video feed) with the selected display elements. Continuing the example above, the display engine 630 may display a cartoon avatar of a person with a martini over a live image feed of the bar. A portion of a live image feed (e.g. an image, a video sequence) with the overlaid display elements may be stored or otherwise published on a social network as an ephemeral message 502 (e.g., via the annotation system 206), according to some example embodiments.

[0080] FIG. 6B shows example functional components of a server venue system 150, according to some example embodiments. As illustrated, the server venue system 150 comprises an interface engine 633 and a places engine 685. The interface engine 633 is configured to receive requests, such as a venue request, from the client device 102. The places engine 635 is configured to use the information sent in a venue request to determine the venues that are nearest to the client device 102. The places engine 635 is further configured to send the set of the nearest venues to the client device 102 as a response to the location request. In some

example embodiments, the set of venues returned to the client device 102 are the five venues that are nearest to the client device 102.

[0081] Further, in some example embodiments, one or more of the engines of the client venue system 210 are integrated into the server venue system 150 as fallbacks. For example, instances of the environment classification engine 610, a venue classification engine 615, an object classification engine 620, a selection engine 625 can be executed from the application server 112 if the client device 102 does not have enough computational power (e.g., processing speed, memory space) to perform classifications and/or selections. In those embodiments, the client device 102 sends image data and location data to the server venue system 150 for processing. The server venue system performs classifications (e.g., the operations of FIG. 7 and below) and sends data back to the client device for further operations (e.g., display of overlay data based upon the venue identified).

[0082] FIG. 7 shows a flow diagram of a method 700 for implementing improved venue selection using machine learning classifications, according to some example embodiments. At operation 705, the location engine 600 transmits a communication to a server (e.g., application server 112) that requests a set of the venues nearest to the client device. The request may include GPS data generated by the client device and other information used to identify nearby venues (e.g., visible IP networks).

[0083] At operation 710, the location engine 600 receives a venue data set from the server. For example, the location engine may receive the five venues that are nearest to the client device 102. At operation 715, the image engine 605 generates one or more images using the client device 102. For example, the image engine 605 uses an image sensor on the client device 102 to generate an image

or video (e.g., multiple images in sequence). In some example embodiments, the client device 102 displays a live video feed on the screen of the client device 102, and one or more images of operation 715 are sampled from the live video stream.

[0084] In some example embodiments, operation 715 occurs before operation 705; that is, the image is generated before the venue request is transmitted. For example, the user may generate an image, and responsive to the image being generated, the client venue system 210 transmits the request of operation 705 to determine where the client device 102 is at the approximate time the image was taken.

[0085] At operation 720, the client venue system 210 classifies the one or more images based on what the images depict. For example, if the image generated at operation 715 is of a coffee mug in a coffee shop, then at operation 720, the client venue system 210 may using a machine learning scheme to identify that the image is of an indoor environment comprising a coffee mug. In some example embodiments, the client venue system 210 implements the machine learning scheme as one or more neural networks. Further details of classification of operations 720 as discussed in detail below with reference to FIG. 8.

[0086] At operation 725, the selection engine 625 selects a venue from the venue data set based on the classification data generated at operation 720. For example, the selection engine 625 may determine that based on the classification data (e.g., an indoor environment, a depicted coffee mug), the coffee shop is the most likely type of venue and thus selects the coffee shop venue type.

[0087] At operation 730, the selection engine 625 selects one or more display objects based on the venue selected. The content selected may be pre-associated in a data structure with the venue categories and sub-categories. At operation 735, the display engine

630 generates a presentation of the image with the selected display objects, according to some example embodiments.

[0088] In some example embodiments, the selection engine 625 passes information about which display objects were selected to the annotation system 206, which can then use the information to generate an ephemeral message 502. At operation 740, the annotation system 206 publishes the presentation to a social network platform as an ephemeral message 502, according to some example embodiments.

[0089] FIG. 8 shows a flow diagram of a method 800 of generating classifications for images, according to some example embodiments. The operations 805-815 of the method 800 may be performed as a subroutine of operation 720 in FIG. 7, according to some example embodiments. At operation 805, the environment classification engine 610 receives an image an input and generates a numerical likelihood that the image of an outside environment and a numerical likelihood that the image is of an inside environment. At operation 810, the venue classification engine 615 receives the image as input and generates numerical likelihoods that the image is of different possible venues. The numerical likelihoods output may depend on how many venues the venue classification engine 615 has been trained on. For example, if the convolutional neural network is trained on images of ten different venues, then at operation 810, the venue classification engine 615 generates ten different numerical likelihoods for a given image.

[0090] At operation 815, the object classification engine 620 generates numerical likelihoods that object depicted in a given image is a known type of object. For example, the object classification engine 620 may first apply a image feature detection scheme (e.g., Sobel edge detection, blob detection, etc.) to find different regions of interest (ROI) within a given image. Each ROI

is a polygon that encircles a given image feature. The image data of each ROI may then be input into the object classification engine 620 for classification. The object classification engine 620 may, for each ROI, generate a plurality of likelihoods that the ROI comprises a given type of object. For example, if the object classification engine 620 is trained on images of five different objects, at operation 815, the object classification engine 620 generates five different numerical likelihoods for each ROI.

[0091] FIG. 9 shows a flow diagram of a method 900 for selecting one or more venues from a venue dataset, according to some example embodiments. At operation 905, the selection engine 625 identifies a set of potential venues (e.g., the venue dataset returned from the application server 112). At operation 910, the selection engine 625 filters out venues in the set of potential venues that do not match the environment classification data generated at operation 805 of FIG. 8. For example, if the environment classification data indicates that a given image is of an outside environment, potential venues that are indoor environments are filtered at operation 910. Venues that have both an outdoor and indoor metadata tag (e.g., a Las Vegas casino modeled after ancient Greece that has a famous indoor casino area and also a famous outdoor pool area) may be not filtered out at operation 910 because a user could be in the outside area or inside area of the venue. At operation 910, the selection engine 625 determines whether the filtering processes of operating 910 has sufficiently narrowed the number of venues. For example, if there are five venues and four are filtered out at operation 910, then then the remaining venue is output as a selection by the selection engine 625 at operation 910. In some embodiments, if a pre-set number of venues are remaining at operation 915, then the remaining venues are output at operation 920 as selections. The

remaining venues (and associated overlay content) can be presented to the user as options and the user can select the correct venue. In some example embodiments, when the selection engine 625 presents different venue options to users, the selection engine 625 records which venue is most often selected by users. Then, when multiple venue options are presented to the user the most often user selected venue is placed at the top of the options list.

[0092] Assuming there are still venues (e.g., more than one venue in the set), the method 900 continues to operation 905. At operation 905, the selection engine 625 determines whether any of the remaining venues have venue classifications generated at operating 810 exceed a pre-specified threshold. If one or more of the remaining venue's venue classification exceeds the pre-specified threshold, the one or more venues are output at operating 930 as selections. For example, if a threshold is set at .90, and an Italian restaurant receives numerical likelihood of .45 and a coffee shop receives a numerical likelihood of .95, the coffee shop is selected at operation 930.

[0093] Further, in some example embodiments, objects recognized by another engine may push a venue's numerical likelihood over the threshold. Each recognized object can have a set weight or value that affects the numerical likelihood of a corresponding venue. For example, assume the threshold is set to 0.90 and the Italian restaurant receives numerical likelihood of .70 and a coffee shop receives a numerical likelihood of .80, thus both are under the threshold and not selected. However, assume later a coffee cup is detected by the object classification engine 620. The recognition of the coffee cup can add 0.2 to the numerical likelihood of the coffee shop thereby pushing the coffee shop over the threshold and causing the coffee shop to be selected by the selection engine 625.

[0094] Assuming at operation 925, none of the venue classifications are strong signals (e.g., signals that exceed the threshold), then the method 900 continues to operation 935. At operation 935, the selection engine 625 determines whether any of the detected objects identified at operation 815 match tags of the remaining venues. If the objects to match a tag, the venues are output at operation 940 as selections. For example, if a coffee mug is detected and the coffee mug is a tag to a food category and a coffee shop sub-category, then any venues in the set that match food or coffee shop are selected at operation 940. On the other hand, if no recognizable objects were detected in the image, or if the detected objects to not match the remaining venues, then the remaining venues are output as selections at operation 945. For example, if three venues remain, the three venues can be displayed as options to the user and the user can select the correct venue.

[0095] Further, as denoted by the dotted arrow extending from operation 935 to 925, the objects analyzed at operation 935 may trigger a readjustment of the venue numerical likelihood (e.g., a recognized object pushes a venue over a threshold).

[0096] In some example embodiments, operation 925 is skipped. For example, after operation 915 the process continues directly to operation 935 for object based selection. In these embodiments, the client venue system is filtering out non-matching environments, then using objects to determine the correct venue.

[0097] Further, in some example embodiments, the selected venues can be further modified to indicate whether they are of the outside portion of the venue, or inside portion of the venue. For example, instead of selecting “casino” as a venue, “casino – outside” is selected. The UI content displayed with the image of the venue can depend on the venue and also whether the user is inside or outside the venue.

[0098] FIG. 10 shows a data structure 1005 that can be used to filter or otherwise select venues and display content, according to some example embodiments. In some example embodiments, the data is stored in a relational database, graph database, and other forms of structured data. The example data structure 1005 comprises categories including a food category 1010, an arts and entertainment category 1015, a shops category 1020, an outdoors category 1025, a travel category 1030, and an education category 1035. Each of the categories may be associated or otherwise linked to subcategories that further refine a given category. For example, the food category 1010 is subdivided into subcategories including a Mexican restaurant food subcategory 1010A, a wine bar subcategory 1010B, and a coffee shop subcategory 1010C. Further, the arts and entertainment category 1015 is subdivided into subcategories including a stadium subcategory 1015A, a cinema subcategory 1015B, and a museum subcategory 1015C. Further, the shops category is further subdivided into subcategories including a gas station subcategory 1020 A, a mall category 1020B, and a high-end shop category 1020C. Further, the outdoors category 1025 is subdivided into subcategories including a beach subcategory 1025A, a park subcategory 1025B, and a lake category 1025C. Further, the travel category 1030 is further subdivided into an airport category 1030A, a train station category 1030B, and a hotel category 1030C. Further, the education category 1035 is subdivided into subcategories including a university subcategory 1035A, a community college subcategory 1035B, and the high school subcategory 1035C. Although a limited number of categories and subcategories are depicted in the example of FIG. 10, it is appreciated by those of ordinary skill in the art that the number of categories and corresponding subcategories can be expanded to include additional categories and subcategories as needed.

[0099] Each of the venues returned as part of the venue dataset is pre-categorized into at least one of the categories or subcategories of the data structure. Each of the categories and sub-categories can be associated with metadata tags that can be used to filter or select categories based on machine learning generated classifications. The data structure and tags can be stored on the local memory of a client device, e.g., client device 102.

[00100] FIG. 11 shows an example of metadata tags with a portion of the data structure, according to some example embodiments. In particular, the food category 1010 is displayed with its associated subcategories including a Mexican restaurant food subcategory 1010A, a wine bar subcategory 1010B, and a coffee shop subcategory 1010C. Each of the categories and subcategories have metadata tags that describe the corresponding data object. The tags can be used for filtering and selecting venues as discussed in FIG. 9. As illustrated, the food category has both the indoor and outdoor tags because food venues can be outdoors, indoors, or both (indoor restaurant with an outdoor patio). Likewise, the subcategories also have environment tags indicating whether the venues are indoor, outdoor, or both. The sub-categories have additional tags that further describe characteristics or objects likely to be detected (e.g., by the object classification engine 620); e.g., the wine bar 1010B has a wine bottle tag and a wine glass tag. When a detected object matches a given tag, the venue may be selected as discussed in FIG. 9.

[00101] Further, each of the categories and sub-categories may have associated or referenced UI content that indicates the category or sub-category venue type. For example, as illustrated, the wine bar sub-category 1010B is linked to UI content 1100 (e.g., avatars, captions) that correspond to the type of associated

category or sub-category. Examples of UI content are shown in FIGS. 12-14.

[00102] FIG. 12 shows an example presentation of an image 1200 of a wine bar with overlay content 1210A and 1210B, according to some example embodiments. In the example of FIG. 12, the client venue system 210 determined that the user (e.g., the human user taking a picture of the wine bar with client device 102) is in an indoor environment and has further detected wine glasses 1205 in the image 1200. The client venue system selects then wine bar subcategory using the above methods 700-900 and data structure 1000. In response to the wine bar selection, associated UI content 1100 is retrieved and overlaid on the image 1200. In particular, as illustrated, the female avatar 1210B of the user with the caption "Wine Time!" 1210A is overlaid on the wine bar image1200. In some example embodiments, the UI content integrated into the image is three-dimensional (e.g., a dancing cartoon hotdog that appears to dance on the wine bar counter top).

[00103] FIG. 13 shows an example presentation of an image 1300 of a casino with overlay content 1305, according to some example embodiments. In the example shown in FIG. 13, the user captured an image of the casino and the client venue system 210 implemented the approaches discussed above to determine that the image is of an indoor environment and further that the image 1300 depicts casino chips and/or a green felt table (e.g., table 1310). Responsive tot he determinations, the client venue system selects an indoor casino environment integrates the overlay content 1305 (a cartoon avatar of the user gambling) onto the image 1300.

[00104] FIG. 14 shows an example image 1400 of a pool area with overlay content 1410 and 1415, according to some example embodiments. In the example of FIG. 14, the client venue system 210 may determine that the user is at the casino, but outside the

building (according to environmental data). Further, the object classification engine 620 identified a pool 1405. Responsive to the determinations, the client venue system 210 selects outdoor content for the casino venue and overlays it on image 1400. In particular, a caption "Vegas Baby!" 1415 and a reclining user avatar 1410 of the user is displayed on image 1400.

[00105] The presentations displayed in FIG. 12-14 can then be published to a social network as an ephemeral message 502, as discussed above. In this way, the venue system disclosed herein implements visual cues, logic, and a specific data structure that allow a client device to quickly and more accurately and quickly determine a current venue, and publish a venue-customized ephemeral message while the user is in or near the venue.

[00106] FIG. 15 is a block diagram illustrating an example of a software architecture that may be installed on a machine, according to some example embodiments. FIG. 15 is merely a non-limiting example of a software architecture, and it will be appreciated that many other architectures may be implemented to facilitate the functionality described herein. The software architecture 1502 may be executing on hardware such as a machine 1600 of FIG. 16 that includes, among other things, processors 1610, memory 1630, and I/O components 1650. A representative hardware layer 1504 is illustrated and can represent, for example, the machine 1600 of FIG. 16. The representative hardware layer 1504 comprises one or more processing units 1506 having associated executable instructions 1508. The executable instructions 1508 represent the executable instructions of the software architecture 1502, including implementation of the methods, modules, and so forth of FIGs. X-

X. The hardware layer 1504 also includes memory or storage modules 1510, which also have the executable instructions 1508. The hardware layer 1504 may also comprise other hardware 1512, which represents any other hardware of the hardware layer 1504, such as the other hardware illustrated as part of the machine 1500.

[00107] In the example architecture of FIG. 15, the software architecture 1502 may be conceptualized as a stack of layers, where each layer provides particular functionality. For example, the software architecture 1502 may include layers such as an operating system 1514, libraries 1516, frameworks/middleware 1518, applications 1520, and a presentation layer 1544. Operationally, the applications 1520 or other components within the layers may invoke API calls 1524 through the software stack and receive a response, returned values, and so forth (illustrated as messages 1526) in response to the API calls 1524. The layers illustrated are representative in nature, and not all software architectures have all layers. For example, some mobile or special purpose operating systems may not provide a frameworks/middleware 1518 layer, while others may provide such a layer. Other software architectures may include additional or different layers.

[00108] The operating system 1514 may manage hardware resources and provide common services. The operating system 1514 may include, for example, a kernel 1528, services 1530, and drivers 1532. The kernel 1528 may act as an abstraction layer between the hardware and the other software layers. For example, the kernel 1528 may be responsible for memory management, processor management (e.g., scheduling), component management, networking, security settings, and so on. The services 1530 may provide other common services for the other software layers. The

drivers 1532 may be responsible for controlling or interfacing with the underlying hardware. For instance, the drivers 1532 may include display drivers, camera drivers, Bluetooth® drivers, flash memory drivers, serial communication drivers (e.g., Universal Serial Bus (USB) drivers), Wi-Fi® drivers, audio drivers, power management drivers, and so forth depending on the hardware configuration.

[00109] The libraries 1516 may provide a common infrastructure that may be utilized by the applications 1520 and/or other components and/or layers. The libraries 1516 typically provide functionality that allows other software modules to perform tasks in an easier fashion than by interfacing directly with the underlying operating system 1514 functionality (e.g., kernel 1528, services 1530, or drivers 1532). The libraries 1516 may include system libraries 1534 (e.g., C standard library) that may provide functions such as memory allocation functions, string manipulation functions, mathematic functions, and the like. In addition, the libraries 1516 may include API libraries 1536 such as media libraries (e.g., libraries to support presentation and manipulation of various media formats such as MPEG4, H.264, MP3, AAC, AMR, JPG, PNG), graphics libraries (e.g., an OpenGL framework that may be used to render 2D and 3D graphic content on a display), database libraries (e.g., SQLite that may provide various relational database functions), web libraries (e.g., WebKit that may provide web browsing functionality), and the like. The libraries 1516 may also include a wide variety of other libraries 1538 to provide many other APIs to the applications 1520 and other software components/modules.

[00110] The frameworks 1518 (also sometimes referred to as middleware) may provide a higher-level common infrastructure that may be utilized by the applications 1520 or other software

components/modules. For example, the frameworks 1518 may provide various graphic user interface (GUI) functions, high-level resource management, high-level location services, and so forth. The frameworks 1518 may provide a broad spectrum of other APIs that may be utilized by the applications 1520 and/or other software components/modules, some of which may be specific to a particular operating system or platform.

[00111] The applications 1520 include built-in applications 1540 and/or third-party applications 1542. Examples of representative built-in applications 1540 may include, but are not limited to, a home application, a contacts application, a browser application, a book reader application, a location application, a media application, a messaging application, or a game application.

[00112] The third-party applications 1542 may include any of the built-in applications 1540, as well as a broad assortment of other applications. In a specific example, the third-party applications 1542 (e.g., an application developed using the Android™ or iOS™ software development kit (SDK) by an entity other than the vendor of the particular platform) may be mobile software running on a mobile operating system such as iOS™, Android™, Windows® Phone, or other mobile operating systems. In this example, the third-party applications 1542 may invoke the API calls 1524 provided by the mobile operating system such as the operating system 1514 to facilitate functionality described herein.

[00113] The applications 1520 may utilize built-in operating system functions (e.g., kernel 1528, services 1530, or drivers 1532), libraries (e.g., system libraries 1534, APIs 1536, and other libraries 1538), or frameworks/middleware 1518 to create user interfaces to interact with users of the system. Alternatively, or additionally, in some systems, interactions with a user may occur

through a presentation layer, such as the presentation layer 1544. In these systems, the application/module "logic" can be separated from the aspects of the application/module that interact with the user.

[00114] Some software architectures utilize virtual machines. In the example of FIG. 15, this is illustrated by a virtual machine 1548. A virtual machine creates a software environment where applications/modules can execute as if they were executing on a hardware machine e.g., the machine 1600 of FIG. 16, for example). A virtual machine 1548 is hosted by a host operating system (e.g., operating system 1514) and typically, although not always, has a virtual machine monitor 1546, which manages the operation of the virtual machine 1548 as well as the interface with the host operating system (e.g., operating system 1514). A software architecture executes within the virtual machine 1548, such as an operating system 1550, libraries 1552, frameworks/middleware 1554, applications 1556, or a presentation layer 1558. These layers of software architecture executing within the virtual machine 1548 can be the same as corresponding layers previously described or may be different.

[00115] FIG. 16 illustrates a diagrammatic representation of a machine 1600 in the form of a computer system within which a set of instructions may be executed for causing the machine to perform any one or more of the methodologies discussed herein, according to an example embodiment. Specifically, FIG. 16 shows a diagrammatic representation of the machine 1600 in the example form of a computer system, within which instructions 1616 (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the machine 1600 to perform any one or more of the methodologies discussed herein may be executed. For example the instructions 1616 may cause the machine 1600 to

execute the method XYZ of FIG. 16. Additionally, or alternatively, the instructions 1616 may implement FIGs. X-X, and so forth. The instructions 1616 transform the general, non-programmed machine 1600 into a particular machine 1600 programmed to carry out the described and illustrated functions in the manner described. In alternative embodiments, the machine 1600 operates as a standalone device or may be coupled (e.g., networked) to other machines. In a networked deployment, the machine 1600 may operate in the capacity of a server machine or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine 1600 may comprise, but not be limited to, a server computer, a client computer, a personal computer (PC), a tablet computer, a laptop computer, a netbook, a set-top box (STB), a PDA, an entertainment media system, a cellular telephone, a smart phone, a mobile device, a wearable device (e.g., a smart watch), a smart home device (e.g., a smart appliance), other smart devices, a web appliance, a network router, a network switch, a network bridge, or any machine capable of executing the instructions 1616, sequentially or otherwise, that specify actions to be taken by the machine 1600. Further, while only a single machine 1600 is illustrated, the term "machine" shall also be taken to include a collection of machines 1600 that individually or jointly execute the instructions 1616 to perform any one or more of the methodologies discussed herein.

[00116] The machine 1600 may include processors 1610, memory 1630, and I/O components 1650, which may be configured to communicate with each other such as via a bus 1602. In an example embodiment, the processors 1610 (e.g., a Central Processing Unit (CPU), a Reduced Instruction Set Computing (RISC) processor, a Complex Instruction Set Computing (CISC)

processor, a Graphics Processing Unit (GPU), a Digital Signal Processor (DSP), an ASIC, a Radio-Frequency Integrated Circuit (RFIC), another processor, or any suitable combination thereof) may include, for example, a processor 1612 and a processor 1614 that may execute the instructions 1616. The term "processor" is intended to include multi-core processors that may comprise two or more independent processors (sometimes referred to as "cores") that may execute instructions contemporaneously. Although FIG. 16 shows multiple processors 1610, the machine 1600 may include a single processor with a single core, a single processor with multiple cores (e.g., a multi-core processor), multiple processors with a single core, multiple processors with multiples cores, or any combination thereof.

[00117] The memory 1630 may include a main memory 1632, a static memory 1634, and a storage unit 1636, both accessible to the processors 1610 such as via the bus 1602. The main memory 1630, the static memory 1634, and storage unit 1636 store the instructions 1616 embodying any one or more of the methodologies or functions described herein. The instructions 1616 may also reside, completely or partially, within the main memory 1632, within the static memory 1634, within the storage unit 1636, within at least one of the processors 1610 (e.g., within the processor's cache memory), or any suitable combination thereof, during execution thereof by the machine 1600.

[00118] The I/O components 1650 may include a wide variety of components to receive input, provide output, produce output, transmit information, exchange information, capture measurements, and so on. The specific I/O components 1650 that are included in a particular machine will depend on the type of machine. For example, portable machines such as mobile phones will likely include a touch input device or other such input

mechanisms, while a headless server machine will likely not include such a touch input device. It will be appreciated that the I/O components 1650 may include many other components that are not shown in FIG. 16. The I/O components 1650 are grouped according to functionality merely for simplifying the following discussion and the grouping is in no way limiting. In various example embodiments, the I/O components 1650 may include output components 1652 and input components 1654. The output components 1652 may include visual components (e.g., a display such as a plasma display panel (PDP), a light emitting diode (LED) display, a liquid crystal display (LCD), a projector, or a cathode ray tube (CRT)), acoustic components (e.g., speakers), haptic components (e.g., a vibratory motor, resistance mechanisms), other signal generators, and so forth. The input components 1654 may include alphanumeric input components (e.g., a keyboard, a touch screen configured to receive alphanumeric input, a photo-optical keyboard, or other alphanumeric input components), point-based input components (e.g., a mouse, a touchpad, a trackball, a joystick, a motion sensor, or another pointing instrument), tactile input components (e.g., a physical button, a touch screen that provides location and/or force of touches or touch gestures, or other tactile input components), audio input components (e.g., a microphone), and the like.

[00119] In further example embodiments, the I/O components 1650 may include biometric components 1656, motion components 1658, environmental components 1660, or position components 1662, among a wide array of other components. For example, the biometric components 1656 may include components to detect expressions (e.g., hand expressions, facial expressions, vocal expressions, body gestures, or eye tracking), measure biosignals (e.g., blood pressure, heart rate, body temperature, perspiration, or

brain waves), identify a person (e.g., voice identification, retinal identification, facial identification, fingerprint identification, or electroencephalogram-based identification), and the like. The motion components 1658 may include acceleration sensor components (e.g., accelerometer), gravitation sensor components, rotation sensor components (e.g., gyroscope), and so forth. The environmental components 1660 may include, for example, illumination sensor components (e.g., photometer), temperature sensor components (e.g., one or more thermometers that detect ambient temperature), humidity sensor components, pressure sensor components (e.g., barometer), acoustic sensor components (e.g., one or more microphones that detect background noise), proximity sensor components (e.g., infrared sensors that detect nearby objects), gas sensors (e.g., gas detection sensors to detection concentrations of hazardous gases for safety or to measure pollutants in the atmosphere), or other components that may provide indications, measurements, or signals corresponding to a surrounding physical environment. The position components 1662 may include location sensor components (e.g., a GPS receiver component), altitude sensor components (e.g., altimeters or barometers that detect air pressure from which altitude may be derived), orientation sensor components (e.g., magnetometers), and the like.

[00120] Communication may be implemented using a wide variety of technologies. The I/O components 1650 may include communication components 1664 operable to couple the machine 1600 to a network 1680 or devices 1670 via a coupling 1682 and a coupling 1672, respectively. For example, the communication components 1664 may include a network interface component or another suitable device to interface with the network 1680. In further examples, the communication components 1664 may

include wired communication components, wireless communication components, cellular communication components, Near Field Communication (NFC) components, Bluetooth® components (e.g., Bluetooth® Low Energy), Wi-Fi® components, and other communication components to provide communication via other modalities. The devices 1670 may be another machine or any of a wide variety of peripheral devices (e.g., a peripheral device coupled via a USB).

[00121] Moreover, the communication components 1664 may detect identifiers or include components operable to detect identifiers. For example, the communication components 1664 may include Radio Frequency Identification (RFID) tag reader components, NFC smart tag detection components, optical reader components (e.g., an optical sensor to detect one-dimensional bar codes such as Universal Product Code (UPC) bar code, multi-dimensional bar codes such as Quick Response (QR) code, Aztec code, Data Matrix, Dataglyph, MaxiCode, PDF417, Ultra Code, UCC RSS-2D bar code, and other optical codes), or acoustic detection components (e.g., microphones to identify tagged audio signals). In addition, a variety of information may be derived via the communication components 1664, such as location via Internet Protocol (IP) geolocation, location via Wi-Fi® signal triangulation, location via detecting an NFC beacon signal that may indicate a particular location, and so forth.

EXECUTABLE INSTRUCTIONS AND MACHINE STORAGE MEDIUM

[00122] The various memories (i.e., 1630, 1632, 1634, and/or memory of the processor(s) 1610) and/or storage unit 1636 may store one or more sets of instructions and data structures (e.g., software) embodying or utilized by any one or more of the

methodologies or functions described herein. These instructions (e.g., the instructions 1616), when executed by processor(s) 1610, cause various operations to implement the disclosed embodiments.

[00123] As used herein, the terms "machine-storage medium," "device-storage medium," "computer-storage medium" mean the same thing and may be used interchangeably in this disclosure. The terms refer to a single or multiple storage devices and/or media (e.g., a centralized or distributed database, and/or associated caches and servers) that store executable instructions and/or data. The terms shall accordingly be taken to include, but not be limited to, solid-state memories, and optical and magnetic media, including memory internal or external to processors. Specific examples of machine-storage media, computer-storage media and/or device-storage media include non-volatile memory, including by way of example semiconductor memory devices, e.g., erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), FPGA, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The terms "machine-storage media," "computer-storage media," and "device-storage media" specifically exclude carrier waves, modulated data signals, and other such media, at least some of which are covered under the term "signal medium" discussed below.

TRANSMISSION MEDIUM

[00124] In various example embodiments, one or more portions of the network 1680 may be an ad hoc network, an intranet, an extranet, a VPN, a LAN, a WLAN, a WAN, a WWAN, a MAN, the Internet, a portion of the Internet, a portion of the PSTN, a plain old telephone service (POTS) network, a cellular telephone

network, a wireless network, a Wi-Fi® network, another type of network, or a combination of two or more such networks. For example, the network 1680 or a portion of the network 1680 may include a wireless or cellular network, and the coupling 1682 may be a Code Division Multiple Access (CDMA) connection, a Global System for Mobile communications (GSM) connection, or another type of cellular or wireless coupling. In this example, the coupling 1682 may implement any of a variety of types of data transfer technology, such as Single Carrier Radio Transmission Technology (1xRTT), Evolution-Data Optimized (EVDO) technology, General Packet Radio Service (GPRS) technology, Enhanced Data rates for GSM Evolution (EDGE) technology, third Generation Partnership Project (3GPP) including 3G, fourth generation wireless (4G) networks, Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSPA), Worldwide Interoperability for Microwave Access (WiMAX), Long Term Evolution (LTE) standard, others defined by various standard-setting organizations, other long range protocols, or other data transfer technology.

[00125] The instructions 1616 may be transmitted or received over the network 1680 using a transmission medium via a network interface device (e.g., a network interface component included in the communication components 1664) and utilizing any one of a number of well-known transfer protocols (e.g., hypertext transfer protocol (HTTP)). Similarly, the instructions 1616 may be transmitted or received using a transmission medium via the coupling 1672 (e.g., a peer-to-peer coupling) to the devices 1670. The terms “transmission medium” and “signal medium” mean the same thing and may be used interchangeably in this disclosure. The terms “transmission medium” and “signal medium” shall be taken to include any intangible medium that is capable of storing, encoding, or carrying the instructions 1616 for execution by the

machine 1600, and includes digital or analog communications signals or other intangible media to facilitate communication of such software. Hence, the terms "transmission medium" and "signal medium" shall be taken to include any form of modulated data signal, carrier wave, and so forth. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a matter as to encode information in the signal.

GLOSSARY

[00126] "CARRIER SIGNAL" in this context refers to any intangible medium that is capable of storing, encoding, or carrying instructions 1616 for execution by the machine 1600, and includes digital or analog communications signals or other intangible media to facilitate communication of such instructions 1616. Instructions 1616 may be transmitted or received over the network 1680 using a transmission medium via a network interface device and using any one of a number of well-known transfer protocols.

[00127] "CLIENT DEVICE" in this context refers to any machine 1600 that interfaces to a communications network 1680 to obtain resources from one or more server systems or other client devices 102. A client device 102 may be, but is not limited to, a mobile phone, desktop computer, laptop, PDA, smartphone, tablet, ultrabook, netbook, multi-processor system, microprocessor-based or programmable consumer electronics system, game console, set-top box, or any other communication device that a user may use to access a network 1680.

[00128] "COMMUNICATIONS NETWORK" in this context refers to one or more portions of a network 1680 that may be an ad hoc network, an intranet, an extranet, a virtual private network

(VPN), a local area network (LAN), a wireless LAN (WLAN), a wide area network (WAN), a wireless WAN (WWAN), a metropolitan area network (MAN), the Internet, a portion of the Internet, a portion of the Public Switched Telephone Network (PSTN), a plain old telephone service (POTS) network, a cellular telephone network, a wireless network, a Wi-Fi® network, another type of network, or a combination of two or more such networks. For example, a network or a portion of a network 1680 may include a wireless or cellular network and the coupling may be a Code Division Multiple Access (CDMA) connection, a Global System for Mobile communications (GSM) connection, or another type of cellular or wireless coupling. In this example, the coupling may implement any of a variety of types of data transfer technology, such as Single Carrier Radio Transmission Technology (1xRTT), Evolution-Data Optimized (EVDO) technology, General Packet Radio Service (GPRS) technology, Enhanced Data rates for GSM Evolution (EDGE) technology, third Generation Partnership Project (3GPP) including 3G, fourth generation wireless (4G) networks, Universal Mobile Telecommunications System (UMTS), High-Speed Packet Access (HSPA), Worldwide Interoperability for Microwave Access (WiMAX), Long-Term Evolution (LTE) standard, others defined by various standard-setting organizations, other long-range protocols, or other data transfer technology.

[00129] "EMPHEMERAL MESSAGE" in this context refers to a message 400 that is accessible for a time-limited duration. An ephemeral message 502 may be a text, an image, a video, and the like. The access time for the ephemeral message 502 may be set by the message sender. Alternatively, the access time may be a default setting or a setting specified by the recipient. Regardless of the setting technique, the message 400 is transitory.

[00130] "MACHINE-READABLE MEDIUM" in this context refers to a component, a device, or other tangible media able to store instructions 1616 and data temporarily or permanently and may include, but is not limited to, random-access memory (RAM), read-only memory (ROM), buffer memory, flash memory, optical media, magnetic media, cache memory, other types of storage (e.g., erasable programmable read-only memory (EPROM)), and/or any suitable combination thereof. The term "machine-readable medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, or associated caches and servers) able to store instructions 1616. The term "machine-readable medium" shall also be taken to include any medium, or combination of multiple media, that is capable of storing instructions 1616 (e.g., code) for execution by a machine 1600, such that the instructions 1616, when executed by one or more processors 1610 of the machine 1600, cause the machine 1600 to perform any one or more of the methodologies described herein. Accordingly, a "machine-readable medium" refers to a single storage apparatus or device, as well as "cloud-based" storage systems or storage networks that include multiple storage apparatus or devices. The term "machine-readable medium" excludes signals per se.

[00131] "COMPONENT" in this context refers to a device, a physical entity, or logic having boundaries defined by function or subroutine calls, branch points, APIs, or other technologies that provide for the partitioning or modularization of particular processing or control functions. Components may be combined via their interfaces with other components to carry out a machine process. A component may be a packaged functional hardware unit designed for use with other components and a part of a program that usually performs a particular function of related functions.

Components may constitute either software components (e.g., code embodied on a machine-readable medium) or hardware components. A "hardware component" is a tangible unit capable of performing certain operations and may be configured or arranged in a certain physical manner. In various example embodiments, one or more computer systems (e.g., a standalone computer system, a client computer system, or a server computer system) or one or more hardware components of a computer system (e.g., a processor 1612 or a group of processors 1610) may be configured by software (e.g., an application or application portion) as a hardware component that operates to perform certain operations as described herein. A hardware component may also be implemented mechanically, electronically, or any suitable combination thereof. For example, a hardware component may include dedicated circuitry or logic that is permanently configured to perform certain operations. A hardware component may be a special-purpose processor, such as a field-programmable gate array (FPGA) or an application-specific integrated circuit (ASIC). A hardware component may also include programmable logic or circuitry that is temporarily configured by software to perform certain operations. For example, a hardware component may include software executed by a general-purpose processor or other programmable processor. Once configured by such software, hardware components become specific machines (or specific components of a machine 1600) uniquely tailored to perform the configured functions and are no longer general-purpose processors 1610. It will be appreciated that the decision to implement a hardware component mechanically, in dedicated and permanently configured circuitry, or in temporarily configured circuitry (e.g., configured by software) may be driven by cost and time considerations. Accordingly, the phrase "hardware component" (or

"hardware-implemented component") should be understood to encompass a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily configured (e.g., programmed) to operate in a certain manner or to perform certain operations described herein.

[00132] Considering embodiments in which hardware components are temporarily configured (e.g., programmed), each of the hardware components need not be configured or instantiated at any one instance in time. For example, where a hardware component comprises a general-purpose processor 1612 configured by software to become a special-purpose processor, the general-purpose processor 1612 may be configured as respectively different special-purpose processors (e.g., comprising different hardware components) at different times. Software accordingly configures a particular processor 1612 or processors 1610, for example, to constitute a particular hardware component at one instance of time and to constitute a different hardware component at a different instance of time.

[00133] Hardware components can provide information to, and receive information from, other hardware components. Accordingly, the described hardware components may be regarded as being communicatively coupled. Where multiple hardware components exist contemporaneously, communications may be achieved through signal transmission (e.g., over appropriate circuits and buses) between or among two or more of the hardware components. In embodiments in which multiple hardware components are configured or instantiated at different times, communications between or among such hardware components may be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple hardware components have access. For example, one hardware component

may perform an operation and store the output of that operation in a memory device to which it is communicatively coupled. A further hardware component may then, at a later time, access the memory device to retrieve and process the stored output. Hardware components may also initiate communications with input or output devices, and can operate on a resource (e.g., a collection of information).

[00134] The various operations of example methods described herein may be performed, at least partially, by one or more processors 1610 that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors 1610 may constitute processor-implemented components that operate to perform one or more operations or functions described herein. As used herein, "processor-implemented component" refers to a hardware component implemented using one or more processors 1610. Similarly, the methods described herein may be at least partially processor-implemented, with a particular processor 1612 or processors 1610 being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors 1610 or processor-implemented components. Moreover, the one or more processors 1610 may also operate to support performance of the relevant operations in a "cloud computing" environment or as a "software as a service" (SaaS). For example, at least some of the operations may be performed by a group of computers (as examples of machines 1600 including processors 1610), with these operations being accessible via a network 1680 (e.g., the Internet) and via one or more appropriate interfaces (e.g., an API). The performance of certain of the operations may be distributed among the processors 1610, not only residing within a single machine 1600, but deployed

across a number of machines 1600. In some example embodiments, the processors 1610 or processor-implemented components may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other example embodiments, the processors 1610 or processor-implemented components may be distributed across a number of geographic locations.

[00135] "PROCESSOR" in this context refers to any circuit or virtual circuit (a physical circuit emulated by logic executing on an actual processor 1612) that manipulates data values according to control signals (e.g., "commands," "op codes," "machine code," etc.) and which produces corresponding output signals that are applied to operate a machine 1600. A processor may, for example, be a central processing unit (CPU), a reduced instruction set computing (RISC) processor, a complex instruction set computing (CISC) processor, a graphics processing unit (GPU), a digital signal processor (DSP), an ASIC, a radio-frequency integrated circuit (RFIC), or any combination thereof. A processor 1610 may further be a multi-core processor 1610 having two or more independent processors 1612, 1614 (sometimes referred to as "cores") that may execute instructions 1616 contemporaneously.

[00136] "TIMESTAMP" in this context refers to a sequence of characters or encoded information identifying when a certain event occurred, for example giving date and time of day, sometimes accurate to a small fraction of a second.

[00137] The following are example embodiments:

[00138] 1. A method comprising: transmitting, using one or more processors of a client device, a request for venue data to one or more servers, the request comprising geolocation data generated by the client device; receiving, from the one or more servers, a venue dataset comprising a plurality of venues, each venue of the

plurality of venues associated with tags that describe the venue; identifying an image generated by the client device; classifying an object depicted in the image using a machine learning scheme; selecting a venue from the venue dataset based at least in part on the classified object matching a tag associated with the venue; selecting one or more display elements that are pre-associated with the selected venue; and displaying, on the client device, a presentation comprising the one or more display elements and the image.

[00139] 2. The method of example 1, further comprising: determining environment data for the image using an additional machine learning scheme, the environment data describing whether the image is of an outside environment or inside environment.

[00140] 3. The method of example 1-2, wherein the selection of the venue is further based at least in part on the environment data of the image matching a tag associated with the venue.

[00141] 4. The method of claim 1-3, wherein the image is part of an active video feed generated by the client device; and the presentation displays the one or more display elements overlaid on the active video feed.

[00142] 5. The method of example 1-4, wherein the plurality of venues are categorized, on the client device, into categories and subcategories, and the tags are metadata of the categories or subcategories.

[00143] 6. The method of example 1-5, wherein the one or more display elements are categorized into the categories and subcategories, and the one or more display elements are pre-associated with the selected venue in that the one or more display

elements and the selected elements share at least one of a same category or subcategory.

[00144] 7. The method of example 1-6, wherein the one or more display elements include an avatar of a user of the client device.

[00145] 8. The method of example 1-7, wherein the geolocation data includes global positioning system (GPS) data generated by a GPS sensor on the client device.

[00146] 9. The method of example 1-8, wherein the venue dataset is a subset of a larger venue dataset that includes venues that are not near to a location indicated by the geolocation data of the client device.

[00147] 10. The method of example 1-9, wherein the machine learning scheme comprises one or more neural networks trained to classify physical objects using an image set, the image set comprising images of a plurality of physical images.

[00148] 11. The method of example 1-10, wherein the additional machine learning scheme comprises one or more neural networks trained to classify environments as being inside or outside, the one or more neural networks trained on an image set comprising images of a plurality of images of outside and inside environments.

[00149] 12. The method of example 1-11, further comprising: receiving, on the client device, an instruction to store the presentation.

[00150] 13. The method of example 1-12, further comprising: publishing, to a social network site, the presentation as an ephemeral message.

[00151] 14. A system comprising: one or more processors of a machine; and a memory storing instructions that, when executed

by the one or more processors, cause the machine to perform any of the methods of examples 1-13.

[00152] 15. A machine-readable storage medium embodying instructions that, when executed by a machine, cause the machine to perform any of the methods of examples 1-13.

CLAIMS

What is claimed is:

1. A method comprising:

transmitting, using one or more processors of a client device, a request for venue data to one or more servers, the request comprising geolocation data generated by the client device;

receiving, from the one or more servers, a venue dataset comprising a plurality of venues, each venue of the plurality of venues associated with tags that describe the venue;

identifying an image generated by the client device;

classifying an object depicted in the image using a machine learning scheme;

selecting a venue from the venue dataset based at least in part on the classified object matching a tag associated with the venue;

selecting one or more display elements that are pre-associated with the selected venue; and

displaying, on the client device, a presentation comprising the one or more display elements and the image.

2. The method of claim 1, further comprising:

determining environment data for the image using an additional machine learning scheme, the environment data describing whether the image is of an outside environment or inside environment.

3. The method of claim 2, wherein the selection of the venue is further based at least in part on the environment data of the image matching a tag associated with the venue.

4. The method of claim 1, wherein the image is part of an active video feed generated by the client device; and
the presentation displays the one or more display elements overlaid on the active video feed.

5. The method of claim 1, wherein the plurality of venues are categorized, on the client device, into categories and subcategories, and the tags are metadata of the categories or subcategories.

6. The method of claim 5, wherein the one or more display elements are categorized into the categories and subcategories, and the one or more display elements are pre-associated with the selected venue in that the one or more display elements and the selected elements share at least one of a same category or subcategory.

7. The method of claim 1, wherein the one or more display elements include an avatar of a user of the client device.

8. The method of claim 1, wherein the geolocation data includes global positioning system (GPS) data generated by a GPS sensor on the client device.

9. The method of claim 1, wherein the venue dataset is a subset of a larger venue dataset that includes venues that are not near to a location indicated by the geolocation data of the client device.

10. The method of claim 1, wherein the machine learning scheme comprises one or more neural networks trained to classify physical objects using an image set, the image set comprising images of a plurality of physical images.

11. The method of claim 2, wherein the additional machine learning scheme comprises one or more neural networks trained to classify environments as being inside or outside, the one or more neural networks trained on an image set comprising images of a plurality of images of outside and inside environments.

12. The method of claim 1, further comprising:

receiving, on the client device, an instruction to store the presentation.

13. The method of claim 12, further comprising:

publishing, to a social network site, the presentation as an ephemeral message.

14. A system comprising:

one or more processors of a machine; and

a memory storing instructions that, when executed by the one or more processors, cause the machine to perform operations comprising:

transmitting a request for venue data to one or more servers, the request comprising geolocation data generated by a client device;

receiving, from the one or more servers, a venue dataset comprising a plurality of venues, each venue of the plurality of venues associated with tags that describe the venue;

identifying an image generated by the client device;

classifying an object depicted in the image using a machine learning scheme;

selecting a venue from the venue dataset based at least in part on the classified object matching a tag associated with the venue;

selecting one or more display elements that are pre-associated with the selected venue; and

displaying, on the client device, a presentation comprising the one or more display elements and the image.

15. The system of claim 14, the operations further comprising:

determining environment data for the image using an additional machine learning scheme, the environment data describing whether the image is of an outside environment or inside environment.

16. The system of claim 15, wherein the selection of the venue is further based at least in part on the environment data of the image matching a tag associated with the venue.

17. The system of claim 14, wherein the image is part of an active video feed generated by the client device; and

the presentation displays the one or more display elements overlaid on the active video feed.

18. The system of claim 14, wherein the plurality of venues are categorized, on the client device, into categories and subcategories, and the tags are metadata of the categories or subcategories.

19. A machine-readable storage medium embodying instructions that, when executed by a machine, cause the machine to perform operations comprising:

transmitting a request for venue data to one or more servers, the request comprising geolocation data generated by the client device;

receiving, from the one or more servers, a venue dataset comprising a plurality of venues near, each venue of the plurality of venues associated with tags that describe the venue;

identifying an image generated by the client device;

classifying an object depicted in the image using a machine learning scheme;

selecting a venue from the venue dataset based at least in part on the classified object matching a tag associated with the venue;

selecting one or more display elements that are pre-associated with the selected venue; and

displaying, on the client device, a presentation comprising the one or more display elements and the image.

20. The machine-readable storage medium of claim 19, the operations further comprising:

determining environment data for the image using an additional machine learning scheme, the environment data describing whether the image is of an outside environment or inside environment.

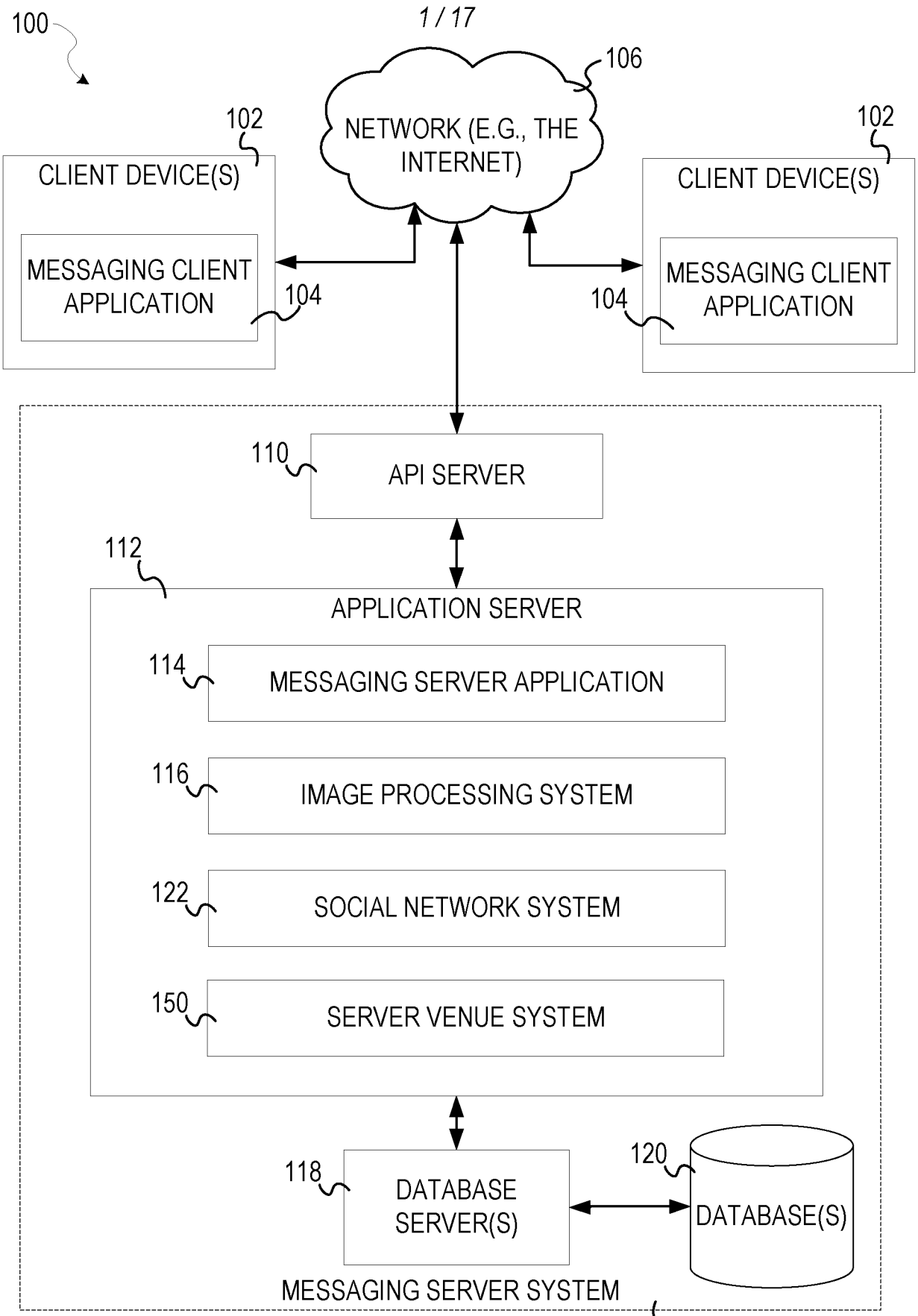

100
106
NETWORK (E.G., THE INTERNET)
102
CLIENT DEVICE(S)
MESSAGING CLIENT APPLICATION
104
102
CLIENT DEVICE(S)
MESSAGING CLIENT APPLICATION
104
110
API SERVER
112
APPLICATION SERVER
114
MESSAGING SERVER APPLICATION
116
IMAGE PROCESSING SYSTEM
122
SOCIAL NETWORK SYSTEM
150
SERVER VENUE SYSTEM
118
DATABASE SERVER(S)
120
DATABASE(S)
MESSAGING SERVER SYSTEM
108

FIG. 1

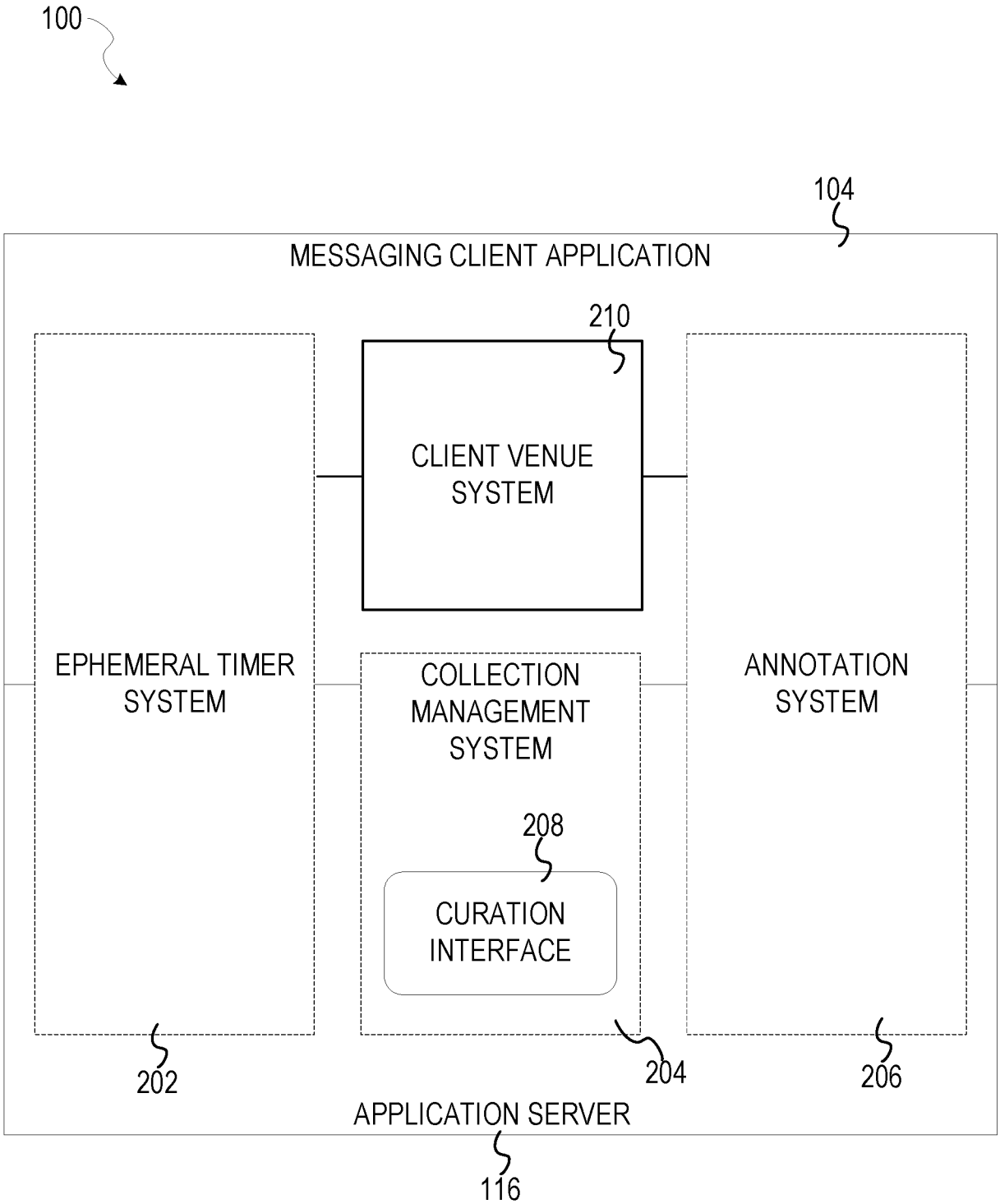

100
104
MESSAGING CLIENT APPLICATION
210
CLIENT VENUE SYSTEM
EPHEMERAL TIMER SYSTEM
COLLECTION MANAGEMENT SYSTEM
208
CURATION INTERFACE
ANNOTATION SYSTEM
202
204
206
APPLICATION SERVER
116

FIG. 2

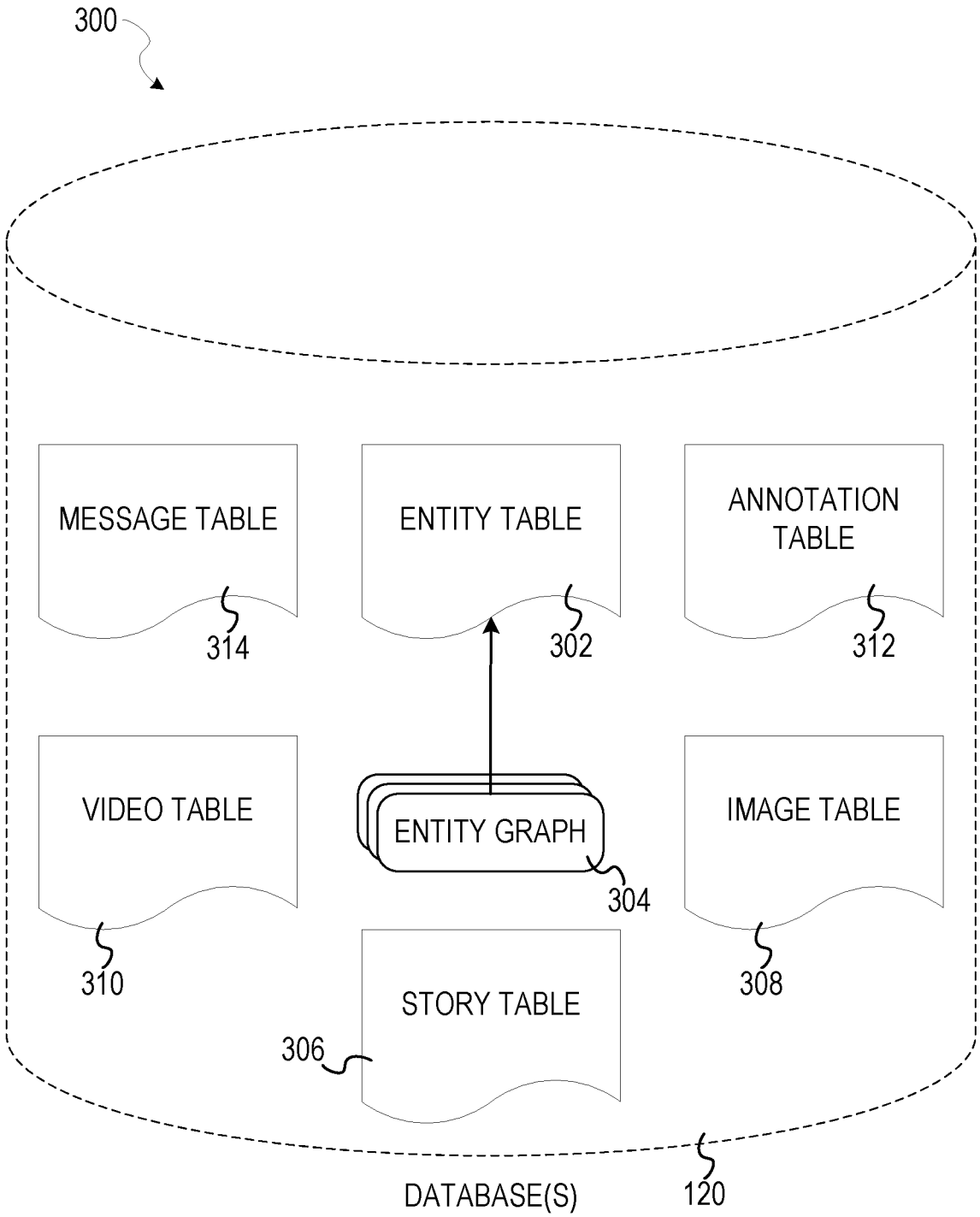

300
MESSAGE TABLE
314
ENTITY TABLE
302
ANNOTATION TABLE
312
VIDEO TABLE
310
ENTITY GRAPH
304
IMAGE TABLE
308
STORY TABLE
306
DATABASE(S)
120

FIG. 3

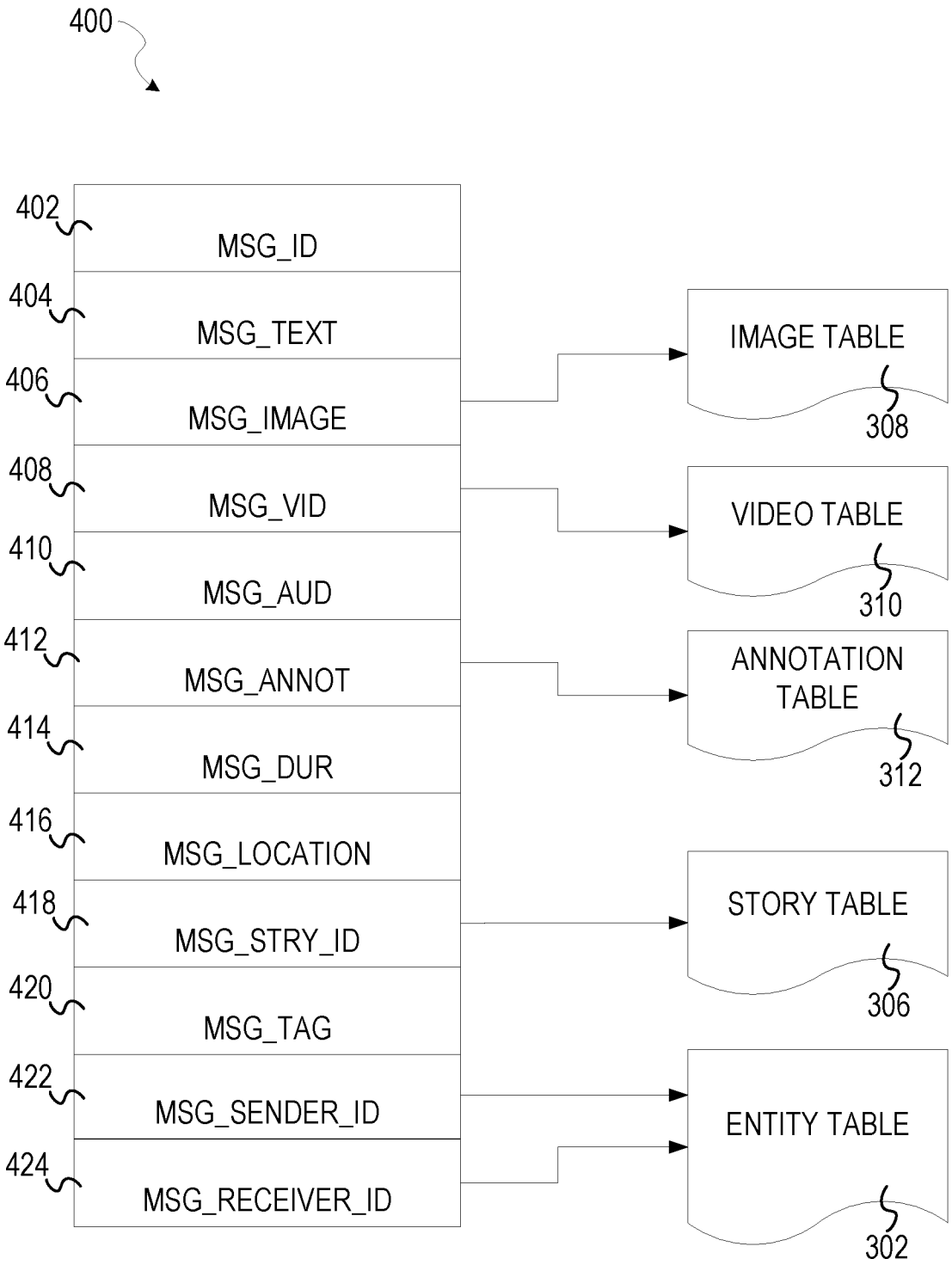

400
402
MSG_ID
404
MSG_TEXT
406
MSG_IMAGE
408
MSG_VID
410
MSG_AUD
412
MSG_ANNOT
414
MSG_DUR
416
MSG_LOCATION
418
MSG_STRY_ID
420
MSG_TAG
422
MSG_SENDER_ID
424
MSG_RECEIVER_ID
IMAGE TABLE
308
VIDEO TABLE
310
ANNOTATION TABLE
312
STORY TABLE
306
ENTITY TABLE
302

FIG. 4

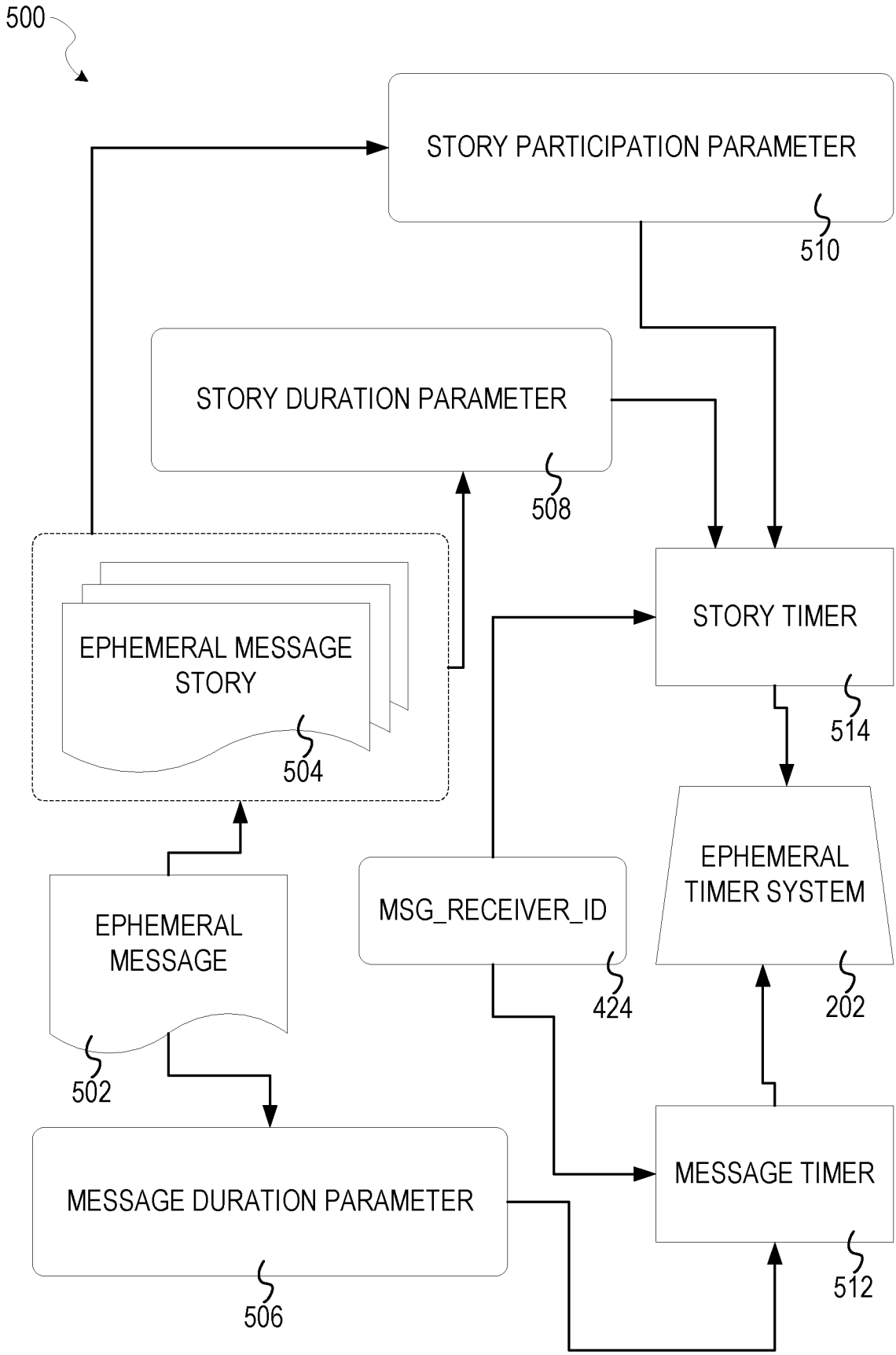

500
STORY PARTICIPATION PARAMETER
510
STORY DURATION PARAMETER
508
EPHEMERAL MESSAGE STORY
504
STORY TIMER
514
EPHEMERAL TIMER SYSTEM
202
MSG_RECEIVER_ID
424
EPHEMERAL MESSAGE
502
MESSAGE DURATION PARAMETER
506
MESSAGE TIMER
512

FIG. 5

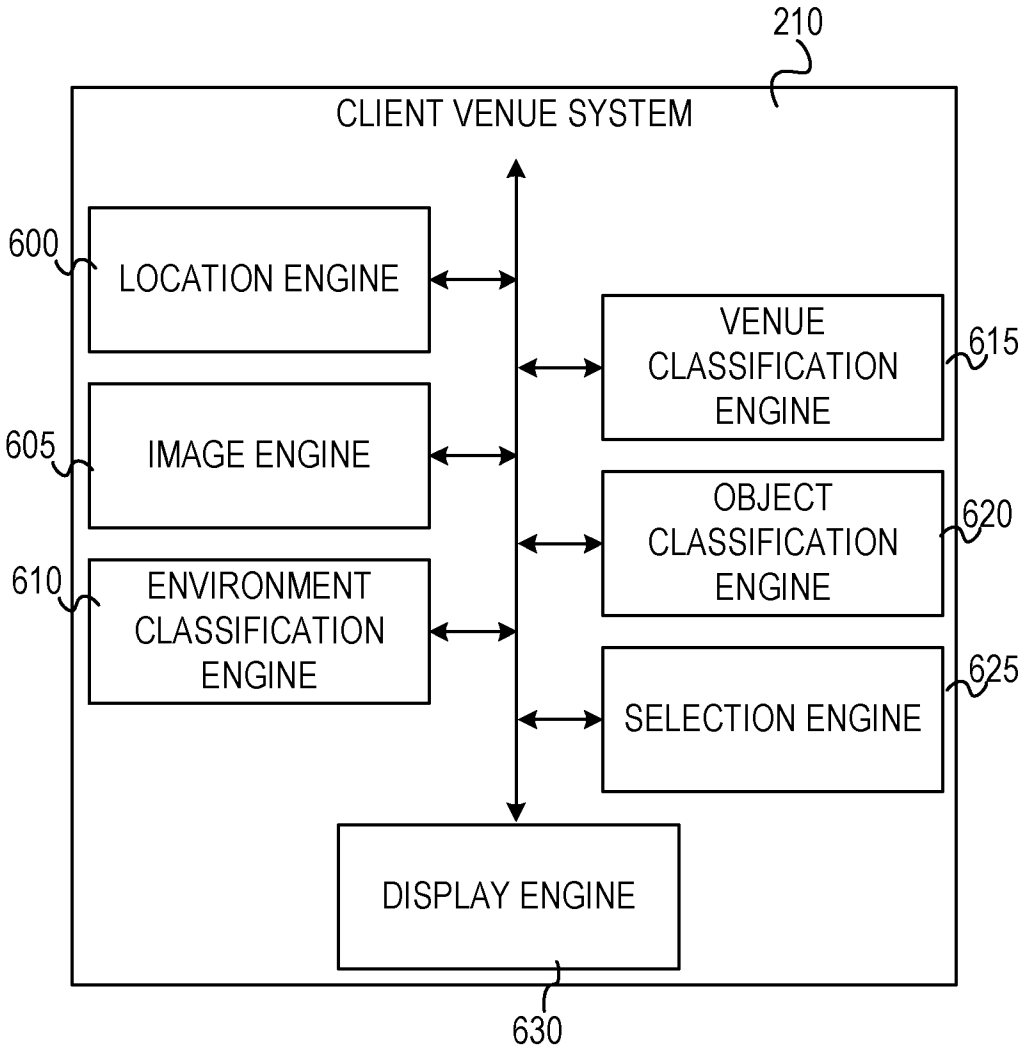

210
CLIENT VENUE SYSTEM
600
LOCATION ENGINE
605
IMAGE ENGINE
610
ENVIRONMENT CLASSIFICATION ENGINE
VENUE CLASSIFICATION ENGINE
615
OBJECT CLASSIFICATION ENGINE
620
SELECTION ENGINE
625
DISPLAY ENGINE
630

FIG. 6A

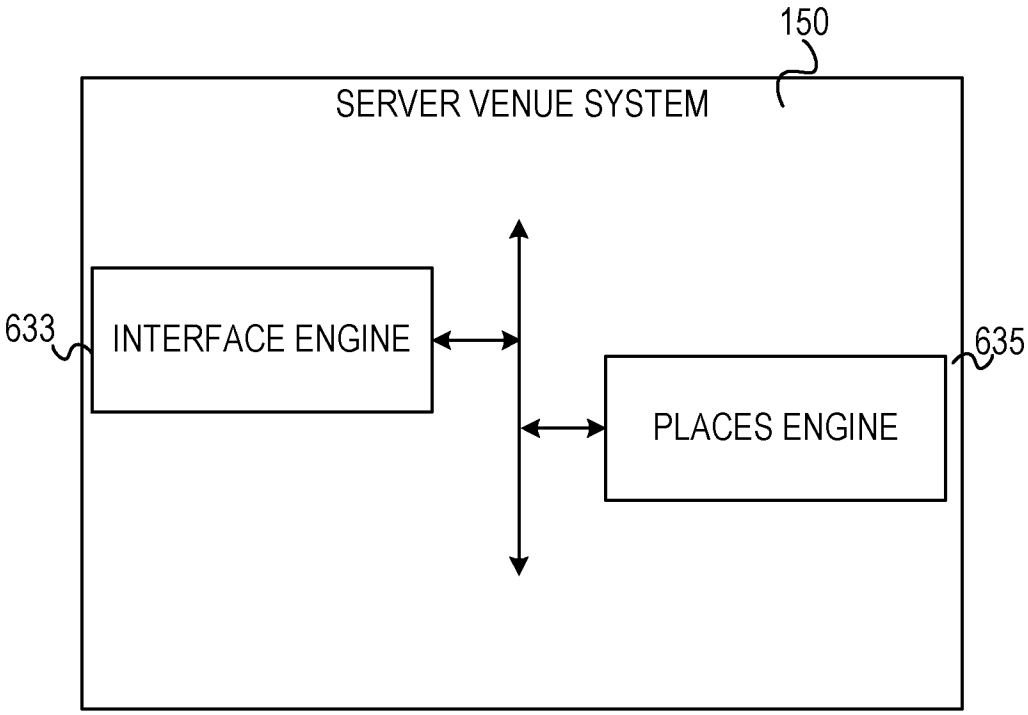

150
SERVER VENUE SYSTEM
633
INTERFACE ENGINE
635
PLACES ENGINE

FIG. 6B

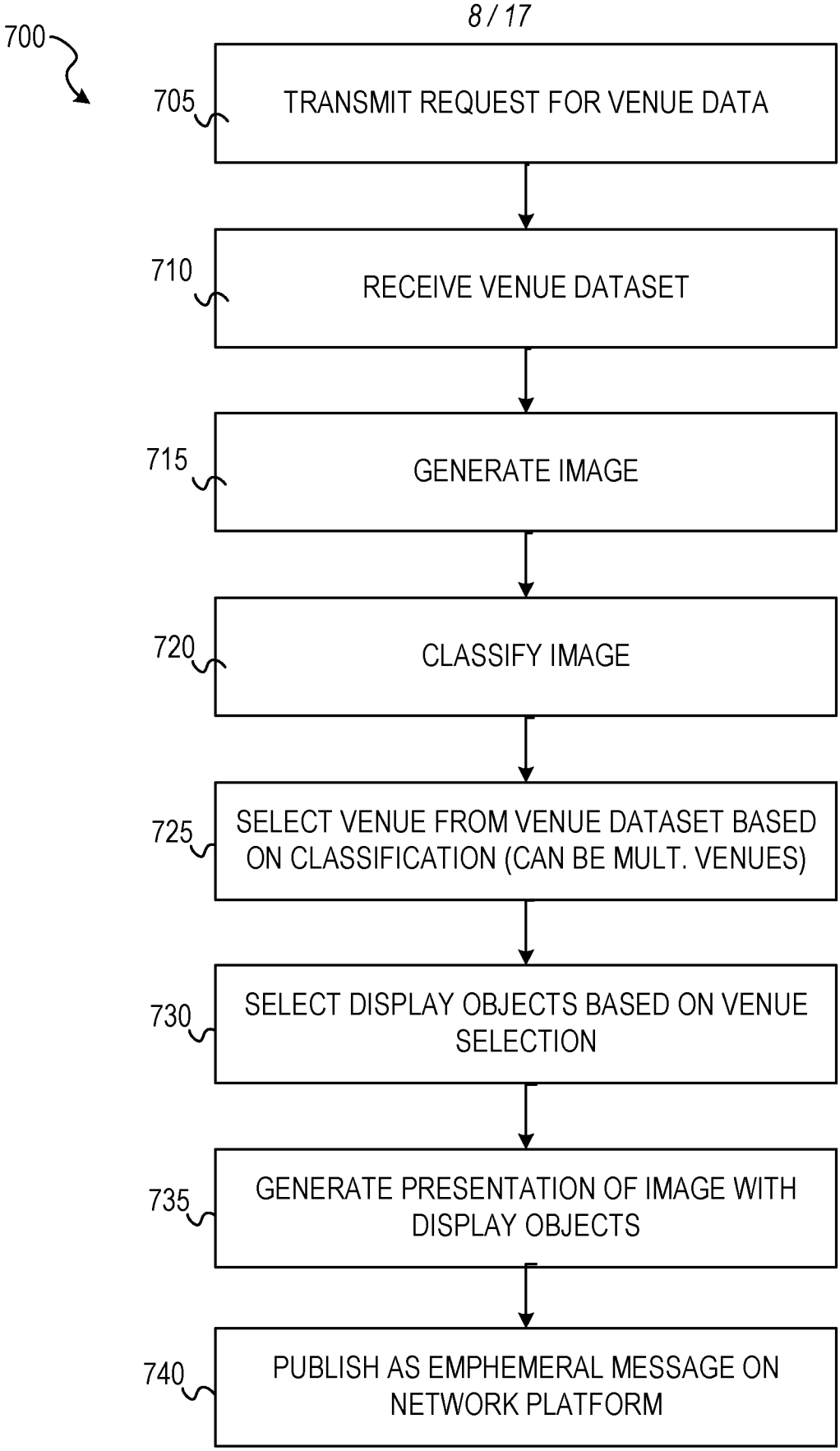

700
705
TRANSMIT REQUEST FOR VENUE DATA
710
RECEIVE VENUE DATASET
715
GENERATE IMAGE
720
CLASSIFY IMAGE
725
SELECT VENUE FROM VENUE DATASET BASED ON CLASSIFICATION (CAN BE MULT. VENUES)
730
SELECT DISPLAY OBJECTS BASED ON VENUE SELECTION
735
GENERATE PRESENTATION OF IMAGE WITH DISPLAY OBJECTS
740
PUBLISH AS EMPHEMERAL MESSAGE ON NETWORK PLATFORM

FIG. 7

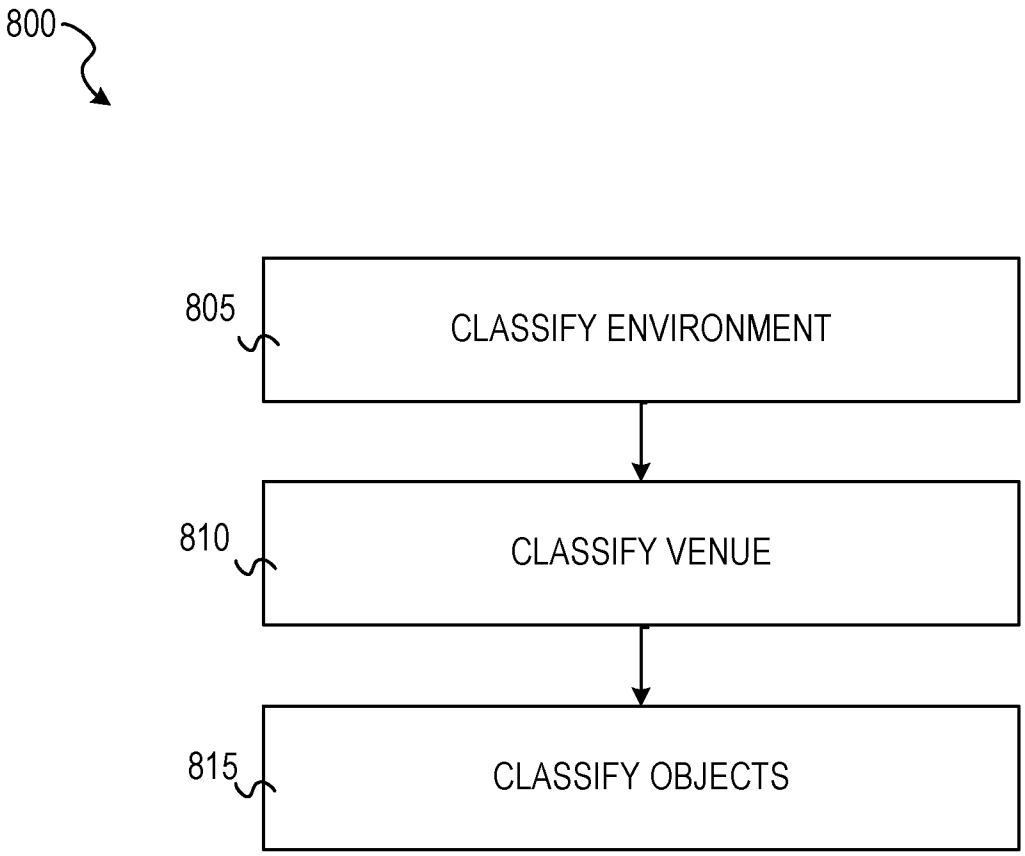

800
805
CLASSIFY ENVIRONMENT
810
CLASSIFY VENUE
815
CLASSIFY OBJECTS

FIG. 8

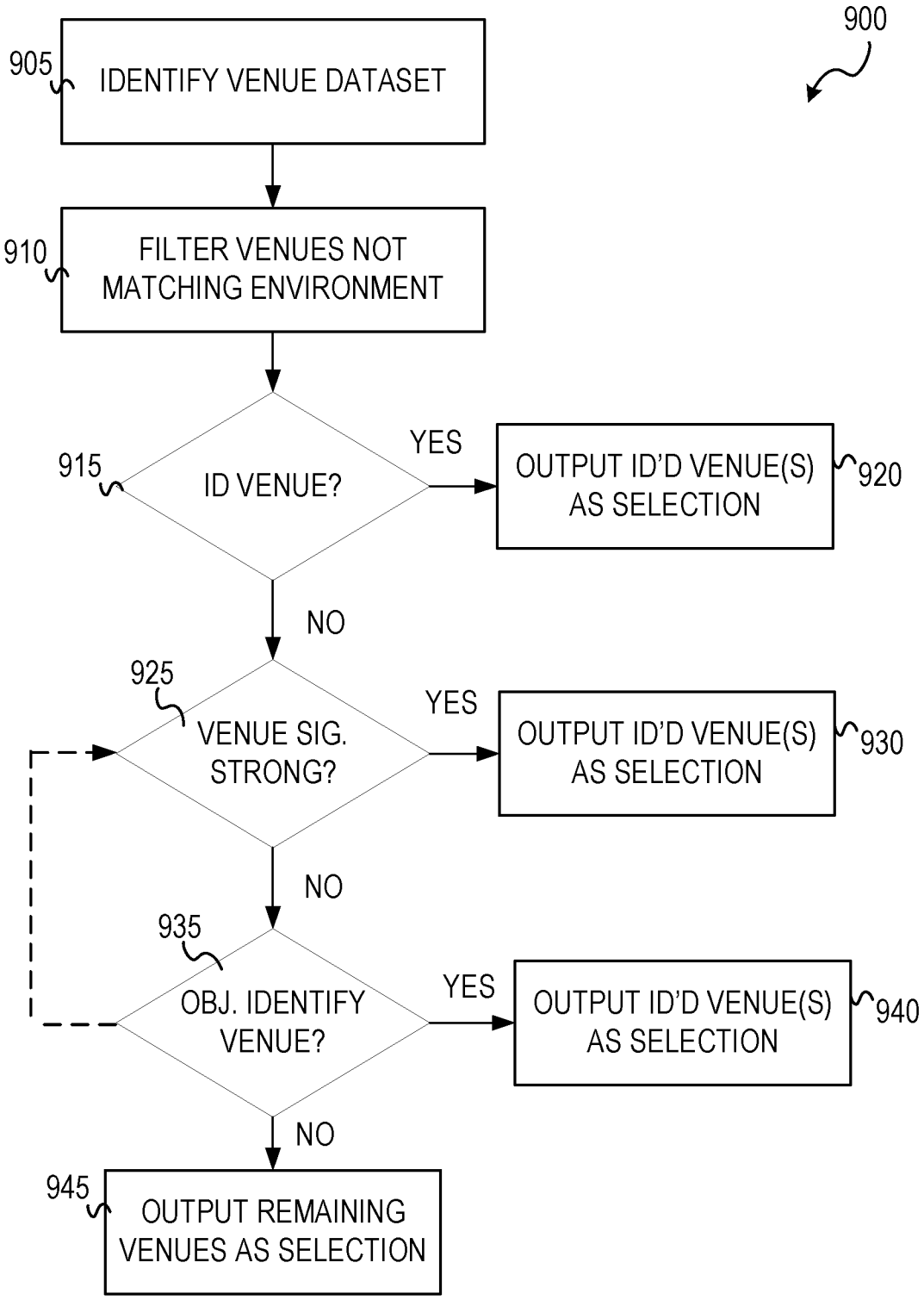

900
905
IDENTIFY VENUE DATASET
910
FILTER VENUES NOT MATCHING ENVIRONMENT
915
ID VENUE?
YES
OUTPUT ID'D VENUE(S) AS SELECTION
920
NO
925
VENUE SIG. STRONG?
YES
OUTPUT ID'D VENUE(S) AS SELECTION
930
NO
935
OBJ. IDENTIFY VENUE?
YES
OUTPUT ID'D VENUE(S) AS SELECTION
940
NO
945
OUTPUT REMAINING VENUES AS SELECTION

FIG. 9

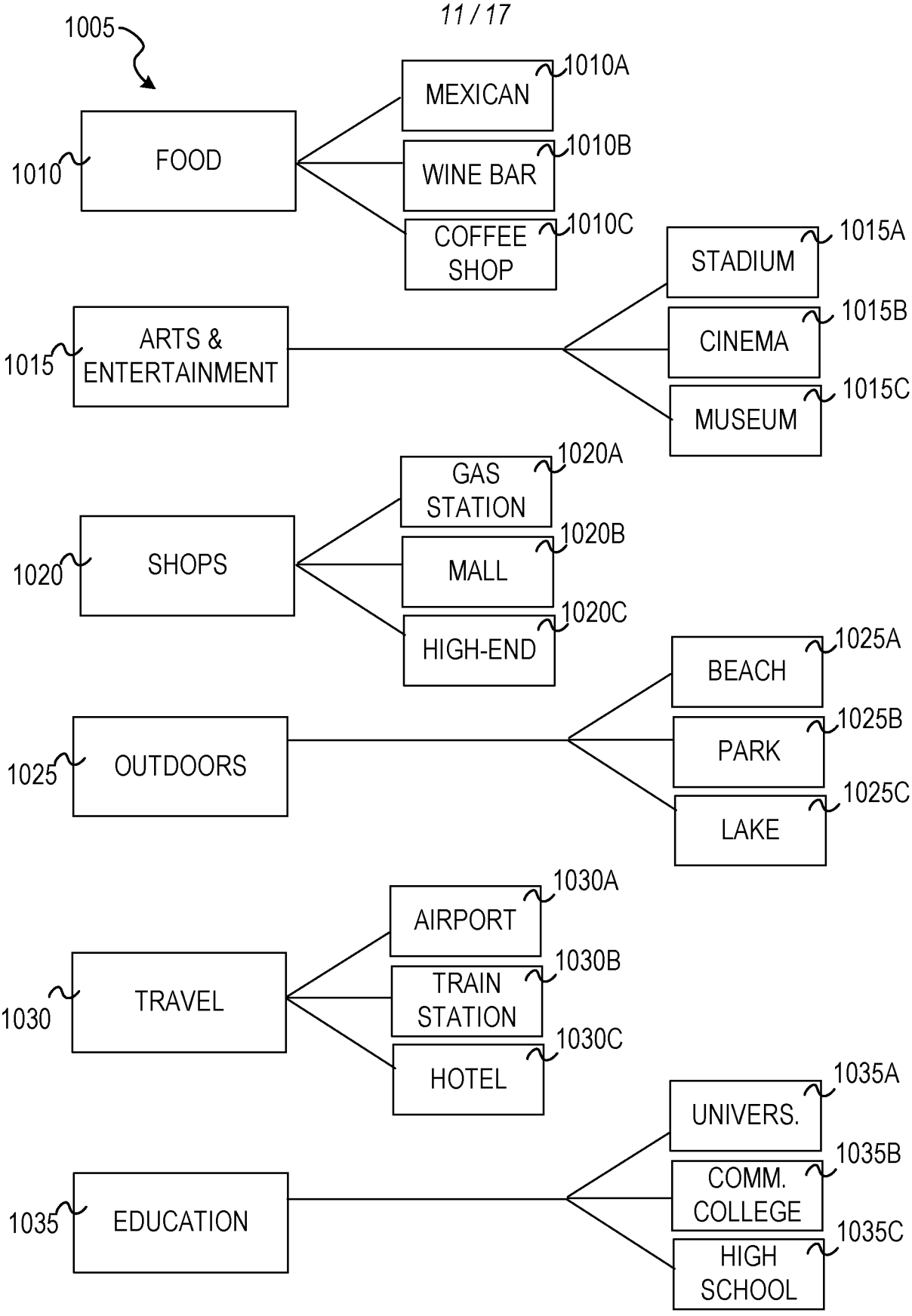

1005
FOOD
1010
MEXICAN
1010A
WINE BAR
1010B
COFFEE SHOP
1010C
ARTS & ENTERTAINMENT
1015
STADIUM
1015A
CINEMA
1015B
MUSEUM
1015C
SHOPS
1020
GAS STATION
1020A
MALL
1020B
HIGH-END
1020C
OUTDOORS
1025
BEACH
1025A
PARK
1025B
LAKE
1025C
TRAVEL
1030
AIRPORT
1030A
TRAIN STATION
1030B
HOTEL
1030C
EDUCATION
1035
UNIVERS.
1035A
COMM. COLLEGE
1035B
HIGH SCHOOL
1035C

FIG. 10

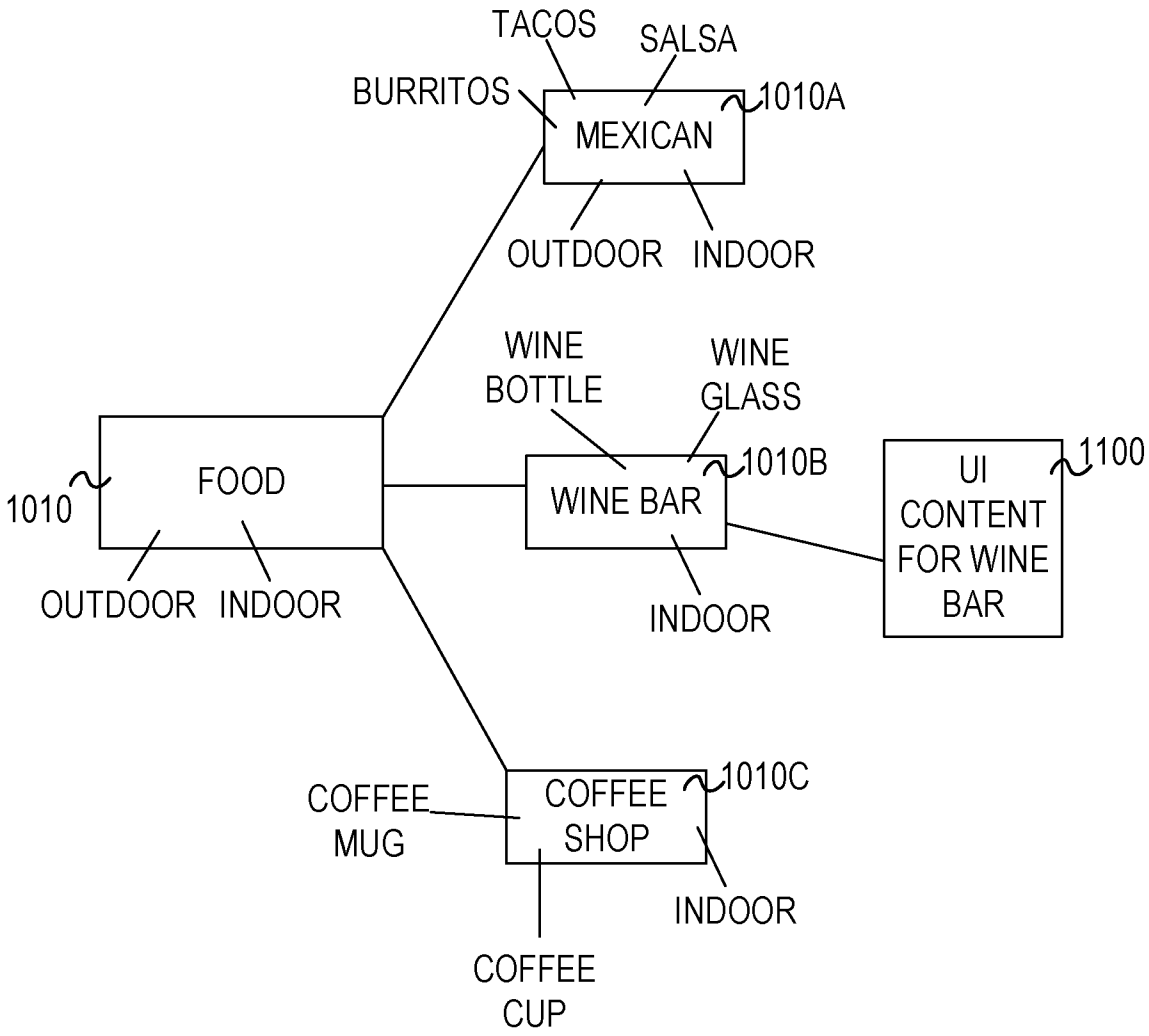

TACOS
SALSA
BURRITOS
MEXICAN
1010A
OUTDOOR
INDOOR
WINE BOTTLE
WINE GLASS
1010
FOOD
WINE BAR
1010B
UI CONTENT FOR WINE BAR
1100
OUTDOOR
INDOOR
INDOOR
COFFEE MUG
COFFEE SHOP
1010C
INDOOR
COFFEE CUP

FIG. 11

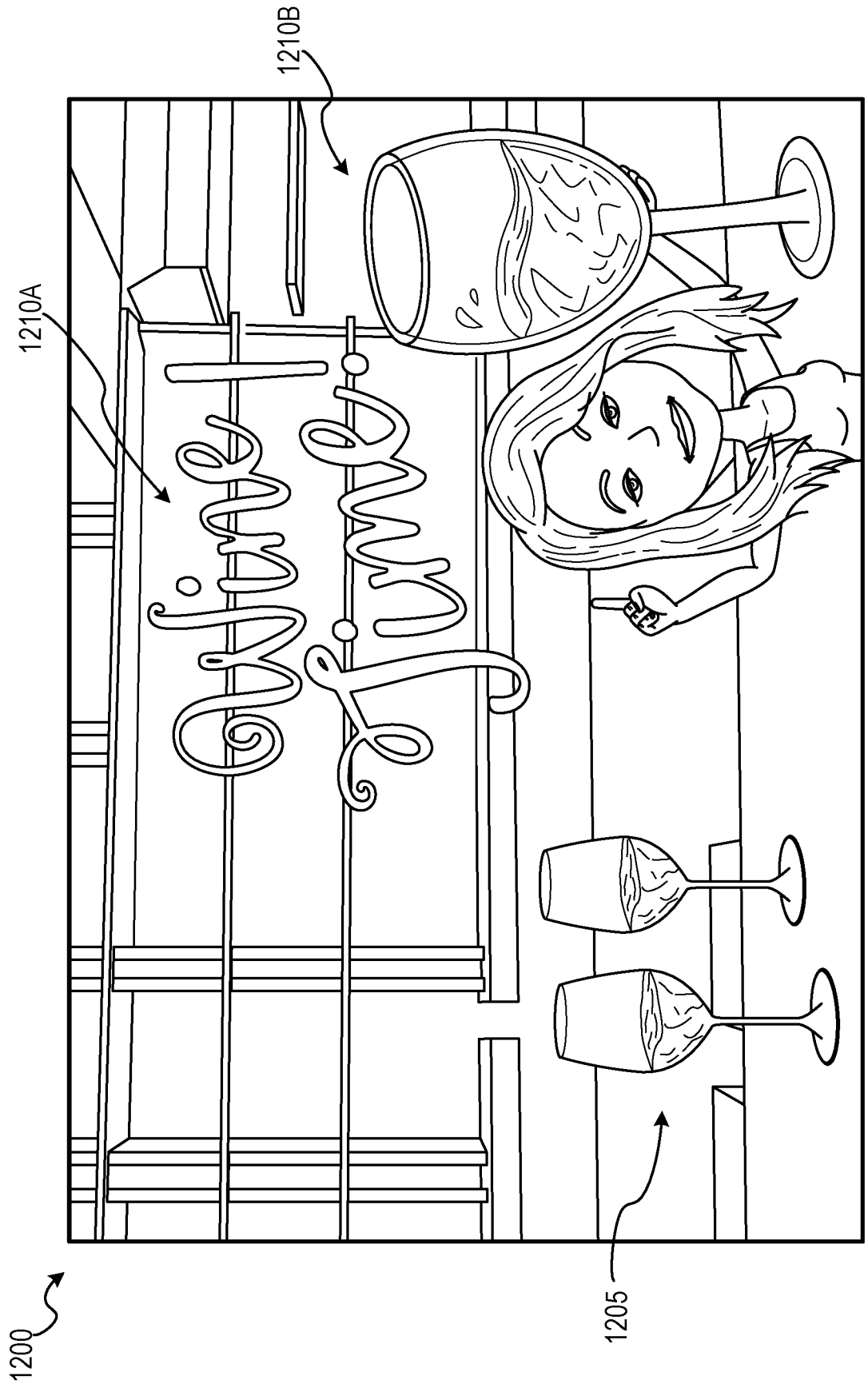


FIG. 12

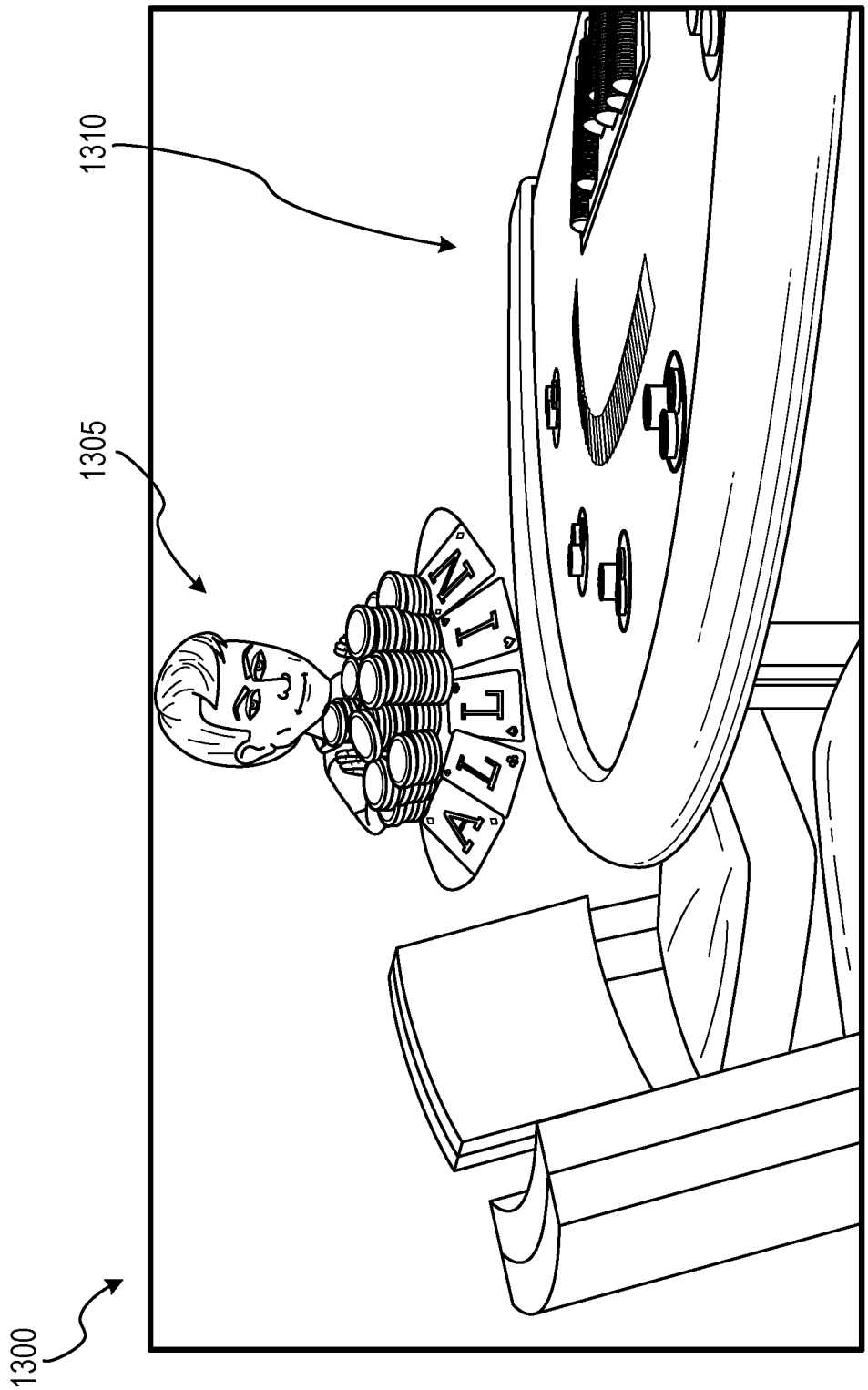


FIG. 13

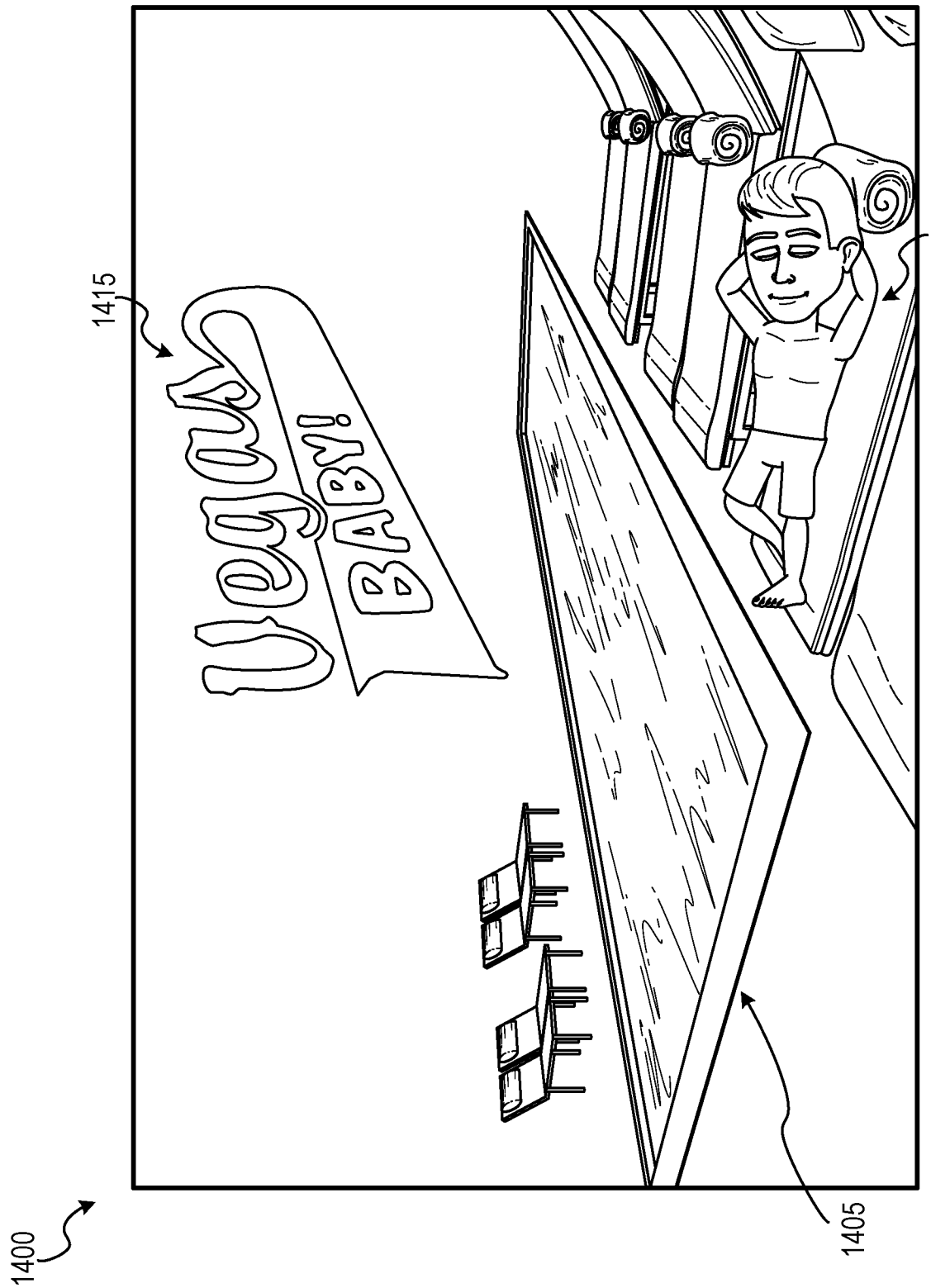


FIG. 14

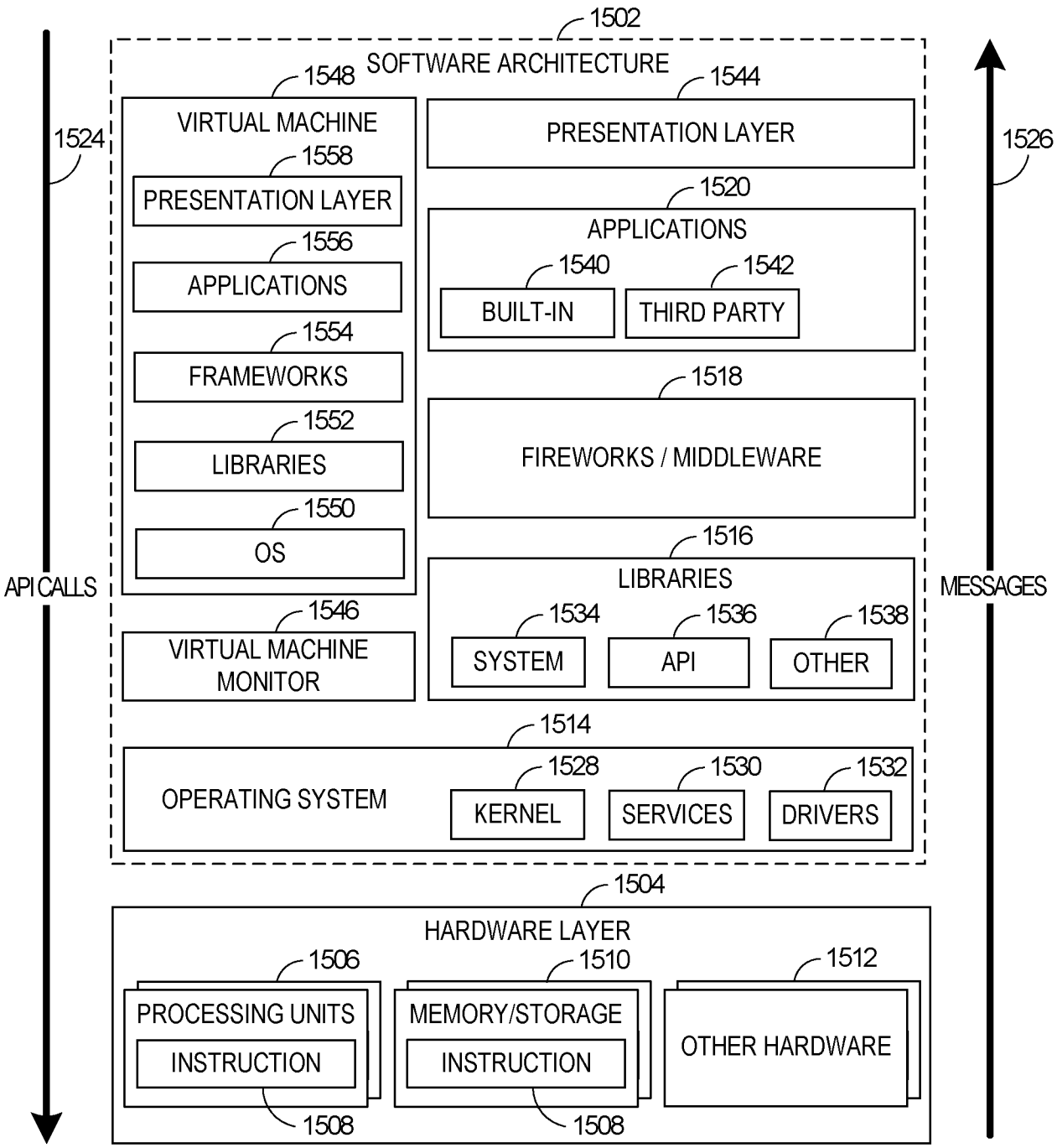

1502
SOFTWARE ARCHITECTURE
1548
VIRTUAL MACHINE
1558
PRESENTATION LAYER
1556
APPLICATIONS
1554
FRAMEWORKS
1552
LIBRARIES
1550
OS
1546
VIRTUAL MACHINE MONITOR
1544
PRESENTATION LAYER
1520
APPLICATIONS
1540
BUILT-IN
1542
THIRD PARTY
1518
FIREWORKS / MIDDLEWARE
1516
LIBRARIES
1534
SYSTEM
1536
API
1538
OTHER
1514
OPERATING SYSTEM
1528
KERNEL
1530
SERVICES
1532
DRIVERS
1504
HARDWARE LAYER
1506
PROCESSING UNITS
INSTRUCTION
1508
1510
MEMORY/STORAGE
INSTRUCTION
1508
1512
OTHER HARDWARE
1524
API CALLS
1526
MESSAGES

FIG. 15

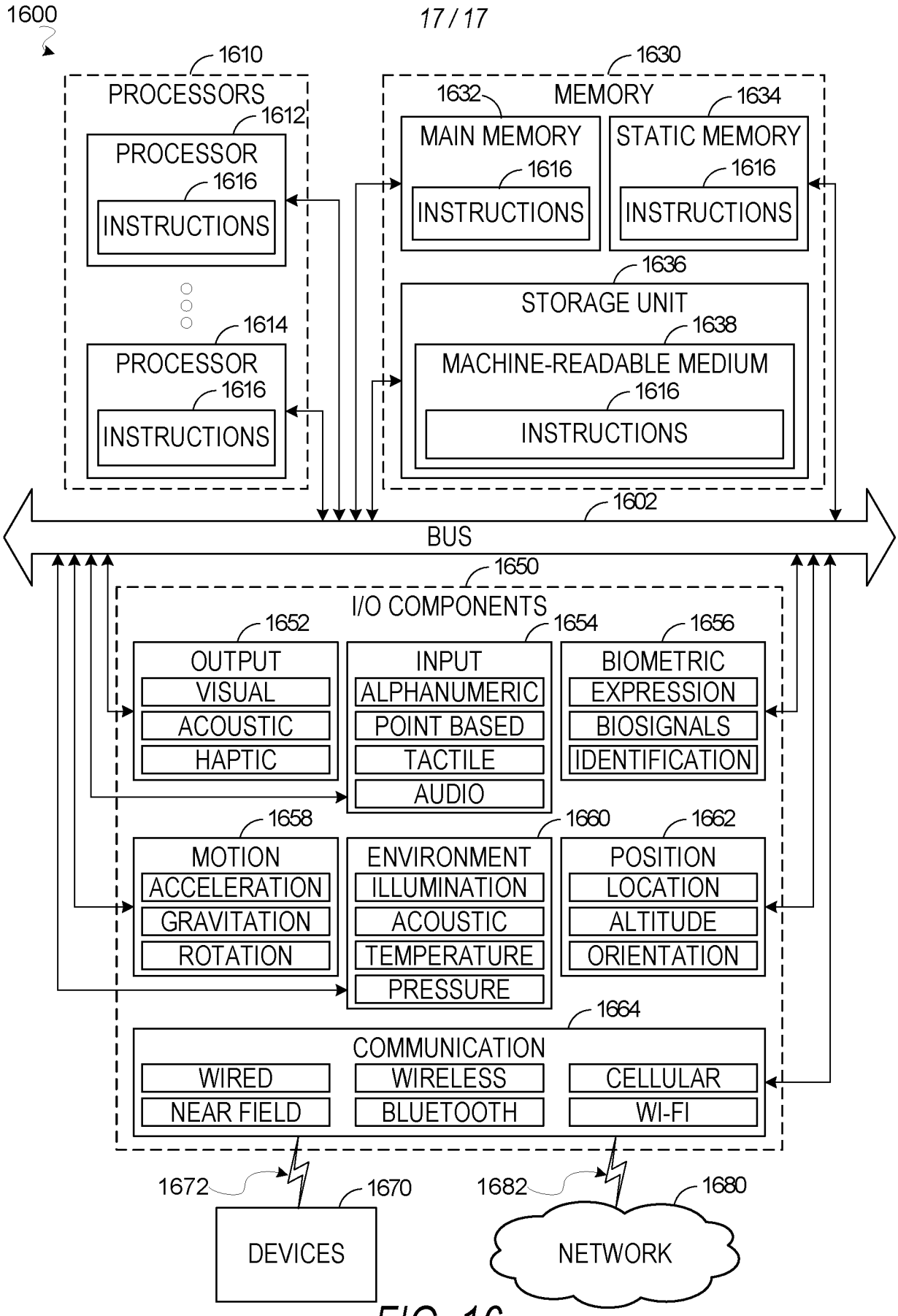

1600
1610
PROCESSORS
1612
PROCESSOR
1616
INSTRUCTIONS
1614
PROCESSOR
1616
INSTRUCTIONS
1630
MEMORY
1632
MAIN MEMORY
1616
INSTRUCTIONS
1634
STATIC MEMORY
1616
INSTRUCTIONS
1636
STORAGE UNIT
1638
MACHINE-READABLE MEDIUM
1616
INSTRUCTIONS
1602
BUS
1650
I/O COMPONENTS
1652
OUTPUT
VISUAL
ACOUSTIC
HAPTIC
1654
INPUT
ALPHANUMERIC
POINT BASED
TACTILE
AUDIO
1656
BIOMETRIC
EXPRESSION
BIOSIGNALS
IDENTIFICATION
1658
MOTION
ACCELERATION
GRAVITATION
ROTATION
1660
ENVIRONMENT
ILLUMINATION
ACOUSTIC
TEMPERATURE
PRESSURE
1662
POSITION
LOCATION
ALTITUDE
ORIENTATION
1664
COMMUNICATION
WIRED
NEAR FIELD
WIRELESS
BLUETOOTH
CELLULAR
WI-FI
1672
1670
DEVICES
1682
1680
NETWORK

FIG. 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2018/049213

A. CLASSIFICATION OF SUBJECT MATTER

H04W 4/33(2018.01)i, G06K 9/00(2006.01)i, G06K 9/62(2006.01)i, G06T 11/60(2006.01)i, H04L 12/58(2006.01)i, H04W 88/02(2009.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04W 4/33; G01C 21/12; G01S 5/00; G06F 17/30; G06K 9/66; G06N 7/00; G06N 99/00; H04W 24/00; H04W 4/02; H04W 8/24; G06K 9/00; G06K 9/62; G06T 11/60; H04L 12/58; H04W 88/02

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: geolocation data, venues, tags, machine learning, object, displaying, environment

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 8774839 B2 (ENHANCED GEOGRAPHIC LLC) 08 July 2014 See column 22, line 39 - column 23, line 24; claims 1, 3, 19; and figure 1.	1-20
A	US 2017-0109615 A1 (GOOGLE INC.) 20 April 2017 See paragraphs [0052]-[0057]; claim 1; and figure 6.	1-20
A	US 2017-0061308 A1 (FUJI XEROX CO., LTD.) 02 March 2017 See paragraphs [0039]-[0046]; claim 1; and figure 1.	1-20
A	KR 10-1115426 B1 (SK PLANET CO., LTD.) 08 March 2012 See paragraphs [0041]-[0058]; claim 1; and figures 2-3.	1-20
A	WO 2017-033141 A1 (GLOPOS FZC) 02 March 2017 See page 27, line 4 - page 29, line 23; claim 1; and figures 4-5.	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

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

Date of the actual completion of the international search	Date of mailing of the international search report
20 December 2018 (20.12.2018)	**26 December 2018 (26.12.2018)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
YANG, Jeong Rok
Telephone No. +82-42-481-5709

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2018/049213

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8774839 B2	08/07/2014	US 2008-0248815 A1	09/10/2008
		US 2012-0302259 A1	29/11/2012
		US 2012-0303446 A1	29/11/2012
		US 2012-0303455 A1	29/11/2012
		US 2013-0073454 A1	21/03/2013
		US 2013-0117109 A1	09/05/2013
		US 2013-0130719 A1	23/05/2013
		US 2013-0137463 A1	30/05/2013
		US 2013-0210462 A1	15/08/2013
		US 2013-0210463 A1	15/08/2013
		US 2014-0195320 A1	10/07/2014
		US 2014-0228056 A1	14/08/2014
		US 2014-0256360 A1	11/09/2014
		US 2014-0278987 A1	18/09/2014
		US 2015-0304819 A1	22/10/2015
		US 2015-0348117 A1	03/12/2015
		US 2017-0061492 A1	02/03/2017
		US 2017-0193553 A1	06/07/2017
		US 8229458 B2	24/07/2012
		US 8364171 B2	29/01/2013
		US 8437776 B2	07/05/2013
		US 8447331 B2	21/05/2013
		US 8515459 B2	20/08/2013
		US 8559977 B2	15/10/2013
		US 8566236 B2	22/10/2013
		US 8626194 B2	07/01/2014
		US 8768379 B2	01/07/2014
		US 8892126 B2	18/11/2014
		US 8996035 B2	31/03/2015
		US 9008691 B2	14/04/2015
		US 9076165 B2	07/07/2015
		US 9277366 B2	01/03/2016
		US 9521524 B2	13/12/2016
		US 9639858 B2	02/05/2017
US 2017-0109615 A1	20/04/2017	WO 2017-066543 A1	20/04/2017
US 2017-0061308 A1	02/03/2017	JP 2017-045435 A	02/03/2017
KR 10-1115426 B1	08/03/2012	KR 10-2006-0072778 A	28/06/2006
WO 2017-033141 A1	02/03/2017	CN 108353248 A	31/07/2018
		EP 3342192 A1	04/07/2018
		US 2017-0064515 A1	02/03/2017
		US 9743253 B2	22/08/2017