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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 April 2016 (28.04.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/065162 A1

(51) International Patent Classification:
G06F 7/00 (2006.01)

(21) International Application Number:
PCT/US2015/056943

(22) International Filing Date:
22 October 2015 (22.10.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/521,278 22 October 2014 (22.10.2014) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 14/521,278 (CON)
Filed on 22 October 2014 (22.10.2014)

(71) Applicant: **PAYPAL, INC.** [US/US]; 2211 N 1st Street, San Jose, California 95131 (US).

(72) Inventor: **ELLIOT, David Edward**; 19565 Canon Drive, Los Gatos, California 95030 (US).

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

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

Published:

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

(54) Title: ON DEMAND GENERATION OF COMPOSITE IMAGES

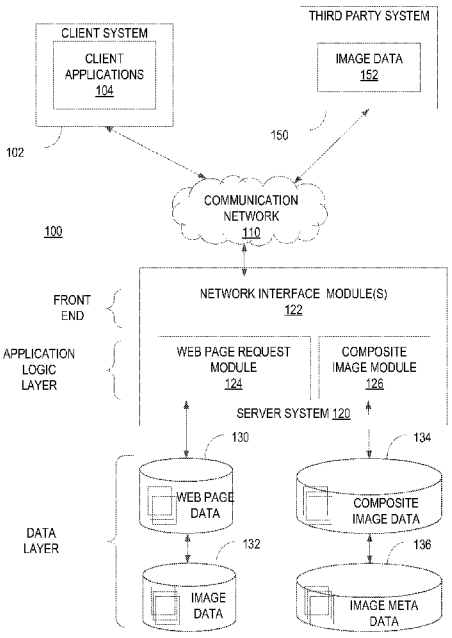

FIG. 1

(57) Abstract: A system and method for improving performance of Web Pages by On- Demand Generation of Composite Images is disclosed. A server system receives a request for a first webpage from a first client system. The server system identifies a list of one or more images referenced in the requested first webpage. The server system groups the identified one or more images into one or more first files. The server system then transmits the one or more first files to the first client system.

WO 2016/065162 A1

ON-DEMAND GENERATION OF COMPOSITE IMAGES

[0001] This International application claims priority to U.S. Application No. 14/521,278, filed October 22, 2014, the entire contents of which are hereby incorporated by reference herein.

TECHNICAL FIELD

[0001] This application relates generally to the field of serving web pages over a network and specifically to improving the efficiency in transmitting data to client systems.

BACKGROUND

[0002] The rise of the computer age has resulted in increased access to personalized services online. As the cost of electronics and networks drop, many services that were previously provided in person are now provided remotely over the Internet. For example, entertainment has increasingly shifted to the online space with companies streaming television (TV) shows and movies to members at home. Similarly, electronic mail (e-mail) has reduced the need for letters to be physically delivered. Instead, messages can be sent over networked systems almost instantly. Online social networking sites allow members to build and maintain personal and business relationships in a much more comprehensive and manageable manner.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The present description is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings, in which:

[0004] FIG. 1 is a block diagram illustrating a client-server system that includes one or more client systems and a server system, in accordance with some embodiments.

[0005] FIG. 2 is a block diagram illustrating a client system, in accordance with some embodiments.

[0006] FIG. 3 is a block diagram illustrating a server system, in accordance with some embodiments.

[0007] FIG. 4A illustrates an exemplary user interface for a web page that displays dynamically generated search results for products available in a networked commerce website in accordance with some embodiments.

[0008] FIG.4B illustrates an exemplary composite image for one or more images associated with a web page in accordance with some embodiments.

[0009] FIG. 5 is a flow diagram illustrating a process for improving performance of web pages by on-demand generation of composite images, in accordance with some embodiments.

[0010] FIGS. 6A-6C are flow diagrams illustrating a process for improving performance of web pages by on-demand generation of composite images in accordance with some embodiments.

[0011] FIG. 7 is a block diagram illustrating an architecture of software, which may be installed on any one or more of devices of a computer system.

[0012] FIG. 8 is a block diagram illustrating components of a machine, according to some example embodiments.

[0013] Like reference numerals refer to corresponding parts throughout the drawings.

DETAILED DESCRIPTION

[0014] Although the embodiments have been described with reference to specific example embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader

scope of the description. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

[0015] In various embodiments, methods and systems for improving performance of web pages by on-demand generation of composite images are disclosed. It is common practice for web pages to include references to images or other files that are not included with the web page and are separately retrieved by the client system. For example, a web page includes an <img> tag with a source file location. The client system requests the image from the source file location and, when it receives the image file, displays it with the rest of the web page. For example, each image results in a separate Hyper Text Transfer Protocol (HTTP) request. However, some web pages include references to a large number of images or other files. Traditionally each image results in a distinct request over a separate web connection. Thus, a web page with a significant number of image requests will result in a large number of connections. This is a problem for two basic reasons. First, if there is a network latency issue that already exists, each separate connection be delayed by it. Secondly, many web browsers limit the number of active connections that a given web page can initiate at a given time. These factors combine to result in a noticeable delay when a web page with a large number of image links is requested.

[0016] This is especially difficult for dynamically generated web pages. The server system cannot pre-generate composite images for web pages that are dynamically generated. This is because the server system does not know which images will be included in the dynamically generated web page (and there are far too many possibilities to pre-generate them all). The server system alleviates this problem by optimizing delivery of image files. A server system receives a web page request from a client system. In response, the server system locates the requested web page and identifies a list of referenced image files (or other files) within the web page. The server system then collects all the referenced image files. All the files are then automatically grouped into a single file. In some example embodiments this single file is a composite image.

[0017] The server system then stores the composite image into a database of composite images. Each composite image includes a list of the component

images and their position within the composite image. The server system then sends the composite image to the requesting client system. The client system then uses information included with the composite images to extract each individual image for display with the requested web page.

[0018] In some example embodiments the server system receives subsequent requests for a web page. Based on the request, the server system generates a list of images referenced in the requested web page. The server system then determines whether the list of requested web pages matches the list associated with any of the stored composite images. In accordance with a determination that the list of requested images for the requested web page match the list of images in a particular composite image, the server system then sends the particular composite image to the requesting client system.

[0019] FIG. 1 is a block diagram illustrating a client-server system 100 that includes one or more client systems 102, a server system 120, and one or more third party systems 150. One or more communication networks 110 interconnect these components. The communication network 110 may be any of a variety of networks, including local area networks (LAN), wide area networks (WAN), wireless networks, wired networks, the Internet, personal area networks (PAN), or a combination of such networks.

[0020] In some example embodiments, a client system 102 is an electronic device, such as a personal computer (PC), a laptop, a smartphone, a tablet, a mobile phone, a wearable computing device, or any other electronic device capable of communication over the communication network 110. Some client systems 102 include one or more client applications 104, which are executed by the client system 102. In some example embodiments, the client application(s) 104 include one or more applications from a set consisting of search applications, communication applications, productivity applications, storage applications, word processing applications, travel applications, or any other useful applications. The client system 102 uses the client application(s) 104 to communicate with the server system 120 and transmit data to, and receive data from, the server system 120.

[0021] In some example embodiments, as shown by way of example in FIG. 1, the server system 120 generally includes three types of components, including

front-end components, application logic components, and data components. As is understood by skilled artisans in the relevant computer and Internet-related arts, each module or engine shown in FIG. 1 represents a set of executable software instructions and the corresponding hardware (e.g., memory and processor) for executing the instructions. To avoid unnecessary detail, various functional modules and engines that are not germane to conveying an understanding of the various example embodiments have been omitted from FIG. 1. However, a skilled artisan will readily recognize that various additional functional modules and engines may be used with a server system 120, such as that illustrated in FIG. 1, to facilitate additional functionality that is not specifically described herein. Furthermore, the various functional modules and engines depicted in FIG. 1 may reside on a single server computer or may be distributed across several server computers in various arrangements. Moreover, although depicted in FIG. 1 as a three component type of architecture, the various example embodiments are by no means limited to this architecture.

[0022] As shown by way of example in FIG. 1, the server system 120 includes network interface module(s) (e.g., a web server) 122, which receives data from various client systems 102, and communicates data back to the appropriate client systems 102 when appropriate. For example, the network interface module(s) 122 receives a web page request from a client system 102 and transmits the web page request to the web page request module 124. The web page request module 124 then accesses the web page database and retrieves and/or generates a web page based on the received web page request. The network interface module(s) 122 then transmits the generated web page to the requesting client system 102.

[0023] As shown by way of example in FIG. 1, the data components include web page data 130, image data 132, composite image data 134, and image metadata 136 for storing data associated with serving web page to client systems. The terms "database," "data," "dataset," and "data storage" are used interchangeably in the specification to refer to data that may or may not be stored in a specific database depending on the exact configuration used in a particular embodiment. Web page data 130 includes any data or image components that the web page request module 124 uses to generate a web page

in response to a web page request. In some example embodiments, the web page request includes a search query and the web page request module 124 searches through the web page data 130 and generates a web page of search results.

[0024] The application logic components include a web page request module 124 and a composite image module 126. The web page request module 124 receives web page requests and in response, generates web pages that are then sent to the requesting client system 102. The composite image module 126 generates composite images from a list of images that are to be sent to client systems 102 in response to web page requests.

[0025] The web page request module 124 receives a web page request from a client system 102. The web page request includes a set of parameters that identify a specific web page or a search query that defines the types of search results the user is interested in. The web page request module 124 then identifies the specific web page or generates a web page based on the web page request. For example, if the user submits a web page request that includes a search query, the web page request module 124 then searches a database to identify one or more search results based on the search query and then generates a web page to display the search results to the user.

[0026] Once the web page has been identified or generated, the web page request module 124 then identifies a list of image files that are referenced in the web page (e.g., with at least one <img> tag). In some example embodiments, the list of image files that are referenced by the web page is generated by the client system 102 and then transmitted to the web page request module 124.

[0027] The composite image module 126 then creates a composite image file. The composite image file includes a list of all the images in the composite image file and data describing where each specific image is within the composite image. Each composite image is stored in a composite image database 134.

[0028] Each time a web page request is received, the web page request module 124 identifies the list of images referenced by the web page. Once the list of images has been identified, the web page request module 124 then determines whether the identified list of images matches the list of images for any of the composite images stored in the composite image database 134. In accordance with a determination that the identified list of images matches the list

of images for a specific composite image, the web page request module 124 then sends the specific composite image to the requesting client system 102 without having to generate a new composite image.

[0029] As shown in FIG. 1, the data layer includes several databases, including databases for storing web page data 130, image data 132, composite image data 134, and image meta data 136.

[0030] In some example embodiments, the web page data 130 includes stored web pages (e.g., pre-generated web pages like a website's homepage), web page components such as templates and graphics that can be used to generate dynamic web pages, and search data that can be used to generate search results (e.g. item data in an e-commerce system).

[0031] In some example embodiments, image data 132 includes one or more images that can be used as components of web pages. Each image can be referenced in a web page that is served by the web page request module 124. The web pages include references which instruct the client system 102 to request the images for display with the web page.

[0032] In some example embodiments, composite image data 134 includes a plurality of already generated composite images. Each composite image includes a plurality of images that have been grouped into a single image file. The composite image module 126 can then store or retrieve files from the composite image data 134. In some example embodiments, the composite image data 134 includes image metadata 136. In some example embodiments, the image metadata 136 is stored in a related but distinct database. The image metadata 136 includes, for each composite image in the composite image data 134, a list of images stored in the associated composite image as well as information identifying where each image is in the composite image (e.g., coordinates within the image file).

[0033] In some example embodiments, third party systems 150 can connect to the server system 120 or the client system 102 through the communication network 110. In some example embodiments, the webpages reference an image stored at a third party system 150 and the server system 120 requests the image from the third party system 150 to use for building a composite image.

[0034] FIG. 2 is a block diagram further illustrating the client system 102, in accordance with some example embodiments. The client system 102 typically includes one or more central processing units (CPUs) 202, one or more network interfaces 210, memory 212, and one or more communication buses 214 for interconnecting these components. The client system 102 includes a user interface 204. The user interface 204 includes a display device 206 and optionally includes an input means such as a keyboard, mouse, a touch sensitive display, or other input buttons 208. Furthermore, the client system 102 may use a microphone and voice recognition to supplement or replace the keyboard as a means of input.

[0035] Memory 212 includes high-speed random access memory, such as DRAM, SRAM, DDR RAM or other random access solid state memory devices, and may include non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory 212 may optionally include one or more storage devices remotely located from the CPU(s) 202. Memory 212, or alternately, the non-volatile memory device(s) within memory 212, comprises a non-transitory computer readable storage medium.

[0036] In some example embodiments, memory 212, or the computer readable storage medium of memory 212, stores the following programs, modules, and data structures, or a subset thereof:

- an operating system 216 that includes procedures for handling various basic system services and for performing hardware dependent tasks;
- a network communication module 218 used for connecting the client system 102 to other computers via the one or more communication network interfaces 210 (wired or wireless) and one or more communication networks (e.g., communication network 110 of FIG. 1), such as the Internet, other WANs, LANs metropolitan area networks (MANs), etc.;
- a display module 220 for enabling the information generated by the operating system 216 to be presented visually as needed;

- one or more client applications 104 for handling various aspects of requesting and receiving numbers, including but not limited to:
    - a web browser module 224 for requesting and displaying data from one or more remote server systems over an network; and
    - a web page analysis module 226 for analyzing the contents of a web page to identify one or more images referenced by the web page that need to be requested to display the web page as intended; and
- client system data module(s) 230 for storing data at the client system 102, including but not limited to:
    - web page data 232 including data that instructs the web browser how to correctly render the web page;
    - image data 234 including data for one or more images to be displayed in the web browser as part of one or more web pages; and
    - user profile data 236 including profile data regarding the user associated with the client system 102 including, but not limited to, demographic information about the user, user interest information, user history information, and any other information regarding the user.

[0037] FIG. 3 is a block diagram illustrating a server system 120, in accordance with some embodiments. The server system 120 typically includes one or more central processing units (CPUs) 302, one or more network interfaces 310, memory 306, and one or more communication buses 308 for interconnecting these components. Memory 306 includes high-speed random access memory, such as DRAM, SRAM, DDR RAM or other random access solid state memory devices; and may include non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory 306 may optionally include one or more storage devices remotely located from the CPU(s) 302.

[0038] Memory 306, or alternately, the non-volatile memory device(s) within memory 306, comprises a non-transitory computer readable storage medium. In some embodiments, memory 306 or the computer readable storage medium of memory 306 stores the following programs, modules, and data structures, or a subset thereof:

- an operating system 314 that includes procedures for handling various basic system services and for performing hardware dependent tasks;
- a network communication module 316 that is used for connecting the server system 120 to other computers via the one or more communication network interfaces 310 (wired or wireless) and one or more communication networks 110, such as the Internet, other WANs, LANS, MANs, and so on;
- one or more server application modules 320 for performing the services offered by server system 120, including but not limited to:
    - a web page request module 124 for receiving web page requests from client systems 102, generating a web page based on the request, and transmitting the generated web page to the requesting client system 102;
    - a composite image module 126 for generating a composite image based on a list of source images;
    - a web page analysis module 326 for parsing, analyzing, and identifying the components of a web page;
    - image list generation module 328 for generating a list of images associated with a web page given web page data 232 that defines the web page;
    - a list comparison module 330 for comparing two lists of images to determine whether they are identical;
    - a composite generation module 334 for generating the layout and composition of a composite image given a list of images;

- an image retrieval module 336 for retrieving images from a third party system (e.g., system 150 in Figure 1) when an image referenced by a web page is not stored in the image data 132 of the server system 120; and

- server data module(s) 340, holding data related to server system 120, including but not limited to:
  - web page data 232 including data used to display a preset or dynamically generated web page at a client system (e.g., system 102 in Figure 1) when rendered by a web browser;
  - image data 132 including images associated with products and services available on an e-commerce site associated with the server system 120;
  - composite image data 134 including one or more composite images created by the server system 120 and metadata associated with each composite image including the list of images in the composite image and the relative positions of each image within the composite image; and
  - user profile data 346 including profile data regarding the user associated with the client system 102 including, but not limited to, demographic information about the user, user interest information, user history information, and any other information regarding the user.

[0039] FIG. 4A illustrates an exemplary user interface 400 for a web page that displays dynamically generated search results for products available in a networked commerce website. In this example, the user interface 400 includes a title bar 402 that displays the name of the website. Below the website title bar 402 is a search field input section 404. This input field allows the user to input one or more search terms as part of a search query. Once the search terms have all been entered, the user selects the search button 406 to initiate the search.

[0040] In this example, a search has already been completed and the web page displays dynamically generated search results 408 below the search input field. The search results include one or more item results (in this case, items 1-6

with number 412, 416, 420, 424, 428, and 432). Each search result includes one or more images that show the item that is associated with the image (410, 414, 418, 422, 426, and 430). Each result also includes item details in the section next to the image.

[0041] FIG.4B illustrates an exemplary composite image for one or more images associated with a web page. In this case, the server system (e.g., server system 120 in Figure 1) analyzes the web page shown in FIG. 4A to determine a list of images that are associated with the web page. The server system 120 the retrieves each image (410, 414, 418, 422, 426, and 430) and creates a composite image 440 that includes all the component images. This composite image 440 can then be transmitted to a client system (e.g., client system 102 in Figure 1) in lieu of sending six separate image files.

[0042] FIG. 5 is a flow diagram illustrating a process for improving performance of web pages by on-demand generation of composite images 440, in accordance with some embodiments. Each of the operations shown in FIG. 5 corresponds, in some embodiments, to instructions stored in a computer memory or computer readable storage medium. In some embodiments, the method 500 described with reference to FIG. 5 is performed by a server system (e.g., server system 120 in FIG. 1).

[0043] The method 500 is performed at a server system (e.g., server system 120 in FIG. 1) including one or more processors and memory 306 storing one or more programs for execution by the one or more processors. The server system (e.g., server system 120 in Figure 1) receives (502) a web page request from a client system (e.g., system 102 in Figure 1). In some example embodiments, the web page request identifies a specific web page (e.g., a predefined webpage). In other embodiments, the web page request includes a search query or other information that will result in the server system (e.g., server system 120 in Figure 1) generating a dynamically generated web page.

[0044] In response to receiving the web page request, the server system (e.g., server system 120 in Figure 1) analyzes (504) the requested web page to identify a list of requested images. The list of requested images include any image (or other file) that is references in the web page data 232 for the requested web page (e.g., in a <img> tag).

[0045] In some example embodiments, the server system (e.g., server system 120 in Figure 1) then determines (506) whether the identified list of requested images match the list of images of any composite images 440 stored in a composite image database 134. The composite image database 134 stores a plurality of previously generated composite images 440 (based on previous web page requests) and stores metadata for each image including, but not limited to a list of images in the composite image 440 and data describing how the images are laid out in the composite image 440.

[0046] In accordance with a determination (508) that the list of requested images matches the list of images included in a composite image 440 in the composite image database 134, the server system (e.g., server system 120 in Figure 1) retrieves (514) the stored matching composite image 440.

[0047] In accordance with a determination (508) that the list of requested images does not match the list of images included in a composite image 440 in the composite image database 134, the server system (e.g., server system 120 in Figure 1) retrieves (510) each image in the list of requested images. In some example embodiments, the images are stored in an image database 132 at the server system (e.g., server system 120 in Figure 1). In other embodiments, the images are stored on a third party server system (e.g., system 150 in Figure 1).

[0048] The server system (e.g., server system 120 in Figure 1) uses the retrieved images to generate (512) a composite image file (e.g., a single file that includes all the retrieved images). The server system (e.g., server system 120 in Figure 1) then sends (516) the composite image file to the requesting client system (e.g., system 102 in Figure 1). The generated composite image can then be twored in the composite image database 134 for future use.

[0049] Figure 6A is a flow diagram illustrating a method for improving performance of web pages by on-demand generation of composite images 440 in accordance with some embodiments. Each of the operations shown in Figure 6A may correspond to instructions stored in a computer memory or computer readable storage medium. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders). In some embodiments, the method described in Figure 6A is performed by the server system (e.g., server system

120 in Figure 1). However, the method described can also be performed by any other suitable configuration of electronic hardware.

[0050] In some embodiments, the method is performed at a server system (e.g., server system 120 in Figure 1) including one or more processors and memory 306 storing one or more programs for execution by the one or more processors.

[0051] In some example embodiments, the server system (e.g., server system 120 in Figure 1) receives (602) a request for a first webpage from a first client system 102. The request is received over a networked computer system (e.g., an HTTP request). In some example embodiments, the request for a web page includes a search query. In some example embodiments, the first webpage is dynamically generated by the server system 120 in response to the request. For example, the webpage request includes a search query and the server system (e.g., server system 120 in Figure 1) dynamically generates a webpage that lists the top search results.

[0052] In some example embodiments, the server system (e.g., server system 120 in Figure 1) identifies (604) a list of one or more images referenced in the requested first webpage. For example, the server system (e.g., server system 120 in Figure 1) parses the data that represents the webpage (e.g., HTML code, Javascript code, PHP scripts, CSS files, and so on) to determine one or more images that the client system (e.g., system 102 in Figure 1) will use when displaying the final webpage to a user. Typically, the client system (e.g., system 102 in Figure 1) sends a unique HTTP request for each image that is requested. The server system (e.g., server system 120 in Figure 1) then creates a list of all the image files (or other files) that are needed to display the webpage.

[0053] In some example embodiments, as part of identifying a list of requested image files, the server system (e.g., server system 120 in Figure 1) transmits (606) the webpage to the first client system (e.g., system 102 in Figure 1). For example, the server system (e.g., server system 120 in Figure 1) transmits the web page data 232 (e.g., HTML data/CSS data) to the client system (e.g., system 102 in Figure 1) for display.

[0054] In some example embodiments, the server system (e.g., server system 120 in Figure 1) receives (608) a list of images requested by the first webpage

from the first client system 102. Each image has a file name and a file location. The server system (e.g., server system 120 in Figure 1) then retrieves all the images in the list of files.

[0055] In some example embodiments, the server system (e.g., server system 120 in Figure 1) then compiles (610) the identified one or more images into one or more first file. In some example embodiments, the one or more first files are composite image files (e.g., large files that include all the images in a set of image files in one file.)

[0056] In some example embodiments, the server system (e.g., server system 120 in Figure 1) has a tool or application that can automatically, given a set of images, create a composite image 440 that includes all the images in the list. This generation of a composite image 440 is only performed in response to a webpage request that includes a dynamically generated webpage because it would be impractical to generate all possible composite images 440.

[0057] In some example embodiments, the server system (e.g., server system 120 in Figure 1) does not include all the images in a webpage in the composite image 440. Instead, the server system (e.g., server system 120 in Figure 1) determines which images to store in the one or more composite images 440 based on a number of factors, including, but not limited to size of the images, color of the image, image location within the page, and so on. In some example embodiments, a webpage is large enough that the entire page cannot be displayed at one time on a given display 206 at a certain level of magnification.

[0058] The server system (e.g., server system 120 in Figure 1) then determines which images will be displayed when the web page is first loaded. Then, the server system (e.g., server system 120 in Figure 1) does not include those images in the composite image 440. Instead the server system (e.g., server system 120 in Figure 1) generates a composite image 440 for all the images on the display that will only be displayed if the user scrolls, clicks a link, or otherwise navigates to view a part of the webpage that was not originally displayed. In this way, the first few images can be displayed without any extra delay while the composite image 440 is being created (although this delay is likely very minimal).

[0059] In some example embodiments, the server system (e.g., server system 120 in Figure 1) first determines whether there is any chance of network latency problems when loading the requested webpage based on information about the size, type, number and location of the requested images. If the chance for network latency problems is small, the server system (e.g., server system 120 in Figure 1) will not create a composite image 440 at all. In some example embodiments, all the decisions concerning whether to make a composite image 440 and what images to group are made automatically by the server system (e.g., server system 120 in Figure 1) without intervention from a user.

[0060] In some example embodiments, the server system (e.g., server system 120 in Figure 1) stores (612) composite images 440 in a database of composite images 134 associated with the server system (e.g., server system 120 in Figure 1). Each composite image 440 includes or has associated metadata that lists a list of images that are included in the composite image 440 and information on which image is displayed in which portion of the composite image 440 for later retrieval.

[0061] The server system (e.g., server system 120 in Figure 1) transmits (614) the first file (e.g., composite image file) to the first requesting client system (e.g., system 102 in Figure 1). This transmitting is done over a network connection.

[0062] In some example embodiments, the server system (e.g., server system 120 in Figure 1) receives (616) a second request for a second webpage from a second client system (e.g., system 102 in Figure 1). In some example embodiments, the first webpage and the second webpage are the same page. For example, two similar users who enter the same search query may receive the same dynamically generated webpage because the search results are the same for both searches.

[0063] Figure 6B is a flow diagram illustrating a method for improving performance of web pages by on-demand generation of composite images 440 in accordance with some embodiments. Each of the operations shown in Figure 6B may correspond to instructions stored in a computer memory or computer readable storage medium. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders). In some embodiments, the method

described in Figure 6B is performed by the server system (e.g., server system 120 in Figure 1). However, the method described can also be performed by any other suitable configuration of electronic hardware.

[0064] In some embodiments, the method is performed at a server system (e.g., server system 120 in Figure 1) including one or more processors and memory storing one or more programs for execution by the one or more processors.

[0065] The server system (e.g., server system 120 in Figure 1) identifies (616) a list of one or more images referenced by the requested second webpage. As described above, the server system (e.g., server system 120 in Figure 1) parses the webpage file data to determine a list of image files that are included when the requested webpage is displayed.

[0066] The server system (e.g., server system 120 in Figure 1) determines (618) whether the list of one or more images matches the list of images for a composite image 440 in the database of composite images 134. In some example embodiments, the server system (e.g., server system 120 in Figure 1) determines (620) whether any of the image files in the list of image files has been updated since the composite image file associated with the first web page was generated.

[0067] In some example embodiments, in accordance with a determination that at least one image in the list of image file has been updated since the composite image file associate with the requested first web page was generated, the server system (e.g., server system 120 in Figure 1) regenerates (628) the composite image file associated with the first web page.

[0068] In some example embodiments, each image file and composite image file has an associated last-updated date. In some example embodiments, the associated last-updated date for each image file and the composite image file are stored in metadata for the composite image file. Further, in some example embodiments, determining whether any of the image files in the list of image files has been updated since the composite image file associated with the first web page was generated, the server system (e.g., server system 120 in Figure 1) determines (622) the last-updated date for the composite image file associated with the first web page.

[0069] In some example embodiments, the server system (e.g., server system 120 in Figure 1) determines (624) whether at least one image file in the list of image files has a last-updated date after the last-updated date for the composite image file. In accordance with a determination that at least one of the image files in the list of image files has a last-updated date after the last-updated date for the composite image file, the server system (e.g., server system 120 in Figure 1) determines (626) that at least one image file in the list of image file has been updated.

[0070] Figure 6C is a flow diagram illustrating a method for improving performance of web pages by on-demand generation of composite images 440 in accordance with some embodiments. Each of the operations shown in Figure 6C may correspond to instructions stored in a computer memory or computer readable storage medium. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders). In some embodiments, the method described in Figure 6C is performed by the server system (e.g., server system 120 in Figure 1). However, the method described can also be performed by any other suitable configuration of electronic hardware.

[0071] In some embodiments, the method is performed at a server system (e.g., server system 120 in Figure 1) including one or more processors and memory storing one or more programs for execution by the one or more processors.

[0072] In accordance with a determination that the list of one or more images matches (630) the list of images for a respective composite image 440 in the database of composite images 134, the server system (e.g., server system 120 in Figure 1) retrieves (632) the respective composite image 440 from the database of composite images 134 and transmits (634) the respective composite image 440 to the second client system.

SOFTWARE ARCHITECTURE

[0073] FIG. 7 is a block diagram illustrating an architecture of software 700, which may be installed on any one or more of devices of FIG. 1 (e.g., client system(s) 102 or server system 120). FIG. 7 is merely a non-limiting example of a software architecture 700 and it will be appreciated that many other

architectures may be implemented to facilitate the functionality described herein. The software 700 may be executing on hardware such as machine 800 of FIG. 8 that includes processors 810, memory 830, and I/O components 850. In the example architecture of FIG. 7, the software 700 may be conceptualized as a stack of layers where each layer may provide particular functionality. For example, the software 700 may include layers such as an operating system 702, libraries 704, frameworks 706, and applications 708. Operationally, the applications 708 may invoke application programming interface (API) calls 710 through the software stack and receive messages 712 in response to the API calls 710.

[0074] The operating system 702 may manage hardware resources and provide common services. The operating system 702 may include, for example, a kernel 720, services 722, and drivers 724. The kernel 720 may act as an abstraction layer between the hardware and the other software layers. For example, the kernel 720 may be responsible for memory management, processor management (e.g., scheduling), component management, networking, security settings, and so on. The services 722 may provide other common services for the other software layers. The drivers 724 may be responsible for controlling and/or interfacing with the underlying hardware. For instance, the drivers 724 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.

[0075] The libraries 704 may provide a low-level common infrastructure that may be utilized by the applications 708. The libraries 704 may include system libraries 730 (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 704 may include API libraries 732 such as media libraries (e.g., libraries to support presentation and manipulation of various media format such as MPREG4, H.264, MP3, AAC, AMR, JPG, PNG), graphics libraries (e.g., an OpenGL framework that may be used to render 2D and 3D in a 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 704 may also include a wide variety of other libraries 734 to provide many other APIs to the applications 708.

[0076] The frameworks 706 may provide a high-level common infrastructure that may be utilized by the applications 708. For example, the frameworks 706 may provide various graphic user interface (GUI) functions, high-level resource management, high-level location services, and so forth. The frameworks 706 may provide a broad spectrum of other APIs that may be utilized by the applications 708, some of which may be specific to a particular operating system 702 or platform.

[0077] The applications 708 include a home application 750, a contacts application 752, a browser application 754, a book reader application 756, a location application 758, a media application 760, a messaging application 762, a game application 764, and a broad assortment of other applications such as third party application 766. In a specific example, the third party application 766 (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 application 766 may invoke the API calls 710 provided by the mobile operating system 702 to facilitate functionality described herein.

EXAMPLE MACHINE ARCHITECTURE AND MACHINE-READABLE MEDIUM

[0078] FIG. 8 is a block diagram illustrating components of a machine 800, 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. Specifically, FIG. 8 shows a diagrammatic representation of the machine 800 in the example form of a computer system, within which instructions 825 (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the machine 800 to perform any one or more of the methodologies discussed

herein may be executed. In alternative embodiments, the machine 800 operates as a standalone device or may be coupled (e.g., networked) to other machines. In a networked deployment, the machine 800 may operate in the capacity of a server machine 120 or a client machine 102 in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine 800 may comprise, but be not limited to, a server computer, a client computer, a PC, a tablet computer, a laptop computer, a netbook, a set-top box (STB), a personal digital assistant (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 825, sequentially or otherwise, that specify actions to be taken by machine 800. Further, while only a single machine 800 is illustrated, the term "machine" shall also be taken to include a collection of machines 800 that individually or jointly execute the instructions 825 to perform any one or more of the methodologies discussed herein.

[0079] The machine 800 may include processors 810, memory 830, and I/O components 850, which may be configured to communicate with each other via a bus 805. In an example embodiment, the processors 810 (e.g., a 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 application specific integrated circuit (ASIC), a radio-frequency integrated circuit (RFIC), another processor, or any suitable combination thereof) may include, for example, processor 815 and processor 820 that may execute instructions 825. The term "processor" is intended to include a multi-core processor 810 that may comprise two or more independent processors (also referred to as "cores") that may execute instructions 825 contemporaneously. Although FIG. 8 shows multiple processors 810, the machine 800 may include a single processor 810 with a single core, a single processor 810 with multiple cores (e.g., a multi-core process), multiple processors 810 with a single core, multiple processors with multiples cores, or any combination thereof.

[0080] The memory 830 may include a main memory 835, a static memory 840, and a storage unit 845 accessible to the processors 810 via the bus 805. The storage unit 845 may include a machine-readable medium 847 on which are stored the instructions 825 embodying any one or more of the methodologies or functions described herein. The instructions 825 may also reside, completely or at least partially, within the main memory 835, within the static memory 840, within at least one of the processors 810 (e.g., within the processor 810's cache memory), or any suitable combination thereof, during execution thereof by the machine 800. Accordingly, the main memory 835, static memory 840, and the processors 810 may be considered as machine-readable media 847.

[0081] As used herein, the term "memory" refers to a machine-readable medium 847 able to store data temporarily or permanently and may be taken to include, but not be limited to, random-access memory (RAM), read-only memory (ROM), buffer memory, flash memory, and cache memory. While the machine-readable medium 847 is shown in an example embodiment to be a single medium, 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 825. The term "machine-readable medium" shall also be taken to include any medium, or combination of multiple media, that is capable of storing instructions (e.g., instructions 825) for execution by a machine (e.g., machine 800), such that the instructions 825, when executed by one or more processors of the machine 800 (e.g., processors 810), cause the machine 800 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" shall accordingly be taken to include, but not be limited to, one or more data repositories in the form of a solid-state memory (e.g., flash memory), an optical medium, a magnetic medium, other non-volatile memory (e.g., erasable programmable read-only memory (EPROM)), or any suitable combination thereof. The term "machine-readable medium" specifically excludes non-statutory signals per se.

[0082] The I/O components 850 may include a wide variety of components to receive input, provide and/or produce output, transmit information, exchange information, capture measurements, and so on. It will be appreciated that the I/O components 850 may include many other components that are not shown in FIG. 8. In various example embodiments, the I/O components 850 may include output components 852 and/or input components 854. The output components 852 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), other signal generators, and so forth. The input components 854 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, and/or other pointing instrument), tactile input components (e.g., a physical button, a touch screen that provide location and force of touches or touch gestures, and/or other tactile input components), audio input components (e.g., a microphone), and the like.

[0083] In further example embodiments, the I/O components 850 may include biometric components 856, motion components 858, environmental components 860, and/or position components 862 among a wide array of other components. For example, the biometric components 856 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, finger print identification, or electroencephalogram based identification), and the like. The motion components 858 may include acceleration sensor components (e.g., accelerometer), gravitation sensor components, rotation sensor components (e.g., gyroscope), and so forth. The environmental components 860 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), and/or other components that may provide indications, measurements, and/or signals corresponding to a surrounding physical environment. The position components 862 may include location sensor components (e.g., a GPS receiver component), altitude sensor components (e.g., altimeters and/or barometers that detect air pressure from which altitude may be derived), orientation sensor components (e.g., magnetometers), and the like.

[0084] Communication may be implemented using a wide variety of technologies. The I/O components 850 may include communication components 864 operable to couple the machine 800 to a network 880 and/or devices 870 via coupling 882 and coupling 872 respectively. For example, the communication components 864 may include a network interface component or other suitable device to interface with the network 880. In further examples, communication components 864 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 870 may be another machine and/or any of a wide variety of peripheral devices (e.g., a peripheral device coupled via a USB).

[0085] Moreover, the communication components 864 may detect identifiers and/or include components operable to detect identifiers. For example, the communication components 864 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, PDF48, Ultra Code, UCC RSS-2D bar code, and other optical codes), acoustic detection components (e.g., microphones to identify tagged audio signals), and so on. In additional, a variety of information may be derived via the communication components 864 such as location via Internet Protocol

(IP) geo-location, location via Wi-Fi® signal triangulation, location via detecting a NFC beacon signal that may indicate a particular location, and so forth.

TRANSMISSION MEDIUM

[0086] In various example embodiments, one or more portions of the network 880 may be an ad hoc network, an intranet, an extranet, a virtual private network (VPN), a LAN, a wireless LAN (WLAN), a 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, the network 880 or a portion of the network 880 may include a wireless or cellular network and the coupling 882 may be a Code Division Multiple Access (CDMA) connection, a Global System for Mobile communications (GSM) connection, or other type of cellular or wireless coupling. In this example, the coupling 882 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.

[0087] The instructions 825 may be transmitted and/or received over the network 880 using a transmission medium via a network interface device (e.g., a network interface component included in the communication components 864) and utilizing any one of a number of well-known transfer protocols (e.g., hypertext transfer protocol (HTTP)). Similarly, the instructions 825 may be transmitted and/or received using a transmission medium via the coupling 872 (e.g., a peer-to-peer coupling) to devices 870. The term “transmission medium”

shall be taken to include any intangible medium that is capable of storing, encoding, or carrying instructions 825 for execution by the machine 800, and includes digital or analog communications signals or other intangible medium to facilitate communication of such software. A transmission medium is one embodiment of a machine-readable medium.

TERM USAGE

[0088] Throughout this specification, plural instances may implement components, operations, or structures described as a single instance. Although individual operations of one or more methods are illustrated and described as separate operations, one or more of the individual operations may be performed concurrently, and nothing requires that the operations be performed in the order illustrated. Structures and functionality presented as separate components in example configurations may be implemented as a combined structure or component. Similarly, structures and functionality presented as a single component may be implemented as separate components. These and other variations, modifications, additions, and improvements fall within the scope of the subject matter herein.

[0089] Although an overview of the inventive subject matter has been described with reference to specific example embodiments, various modifications and changes may be made to these embodiments without departing from the broader scope of embodiments of the present disclosure. Such embodiments of the inventive subject matter may be referred to herein, individually or collectively, by the term "invention" merely for convenience and without intending to voluntarily limit the scope of this application to any single disclosure or inventive concept if more than one is, in fact, disclosed.

[0090] The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived therefrom, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

[0091] As used herein, the term "or" may be construed in either an inclusive or exclusive sense. Moreover, plural instances may be provided for resources, operations, or structures described herein as a single instance. Additionally, boundaries between various resources, operations, modules, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within a scope of various embodiments of the present disclosure. In general, structures and functionality presented as separate resources in the example configurations may be implemented as a combined structure or resource. Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements fall within a scope of embodiments of the present disclosure as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

1. A computer implemented method comprising:

receiving a request for a first webpage from a first client system;

identifying a list of one or more images referenced in the requested first webpage;

compiling the identified one or more images into one or more composite image files; and

transmitting the one or more composite image files to the first client system.

2. The method of claim 1, further comprising storing the one or more composite image files in a database of composite images, wherein each composite image file includes a list of images that are included in the composite image file.

3. The method of claim 2, further including:

receiving a second request for a second webpage from a second client system;

identifying a list of one or more images referenced by the requested second webpage;

determining whether the list of one or more images matches the list of images for a composite image file in the database of composite images;

in accordance with a determination that the list of one or more images matches the list of images for a respective composite image in the database of composite images:

retrieving the respective composite image from the database of composite images; and

transmitting the respective composite image to the second client system.

4. The method of claim 3, wherein the first webpage and the second webpage are the same webpage.

5. The method of claim 1, wherein the first webpage is dynamically generated by the server system.

6. The method of claim 1, wherein the request for the first webpage includes a search query.

7. The method of claim 2, wherein prior to compiling the identified one or more images into one or more composite image files, determining whether the identified list of one or more images matches the list of images for a composite image file in the database of composite images.

8. The method of claim 7, further including:

in accordance with a determination the identified list of one or more images matches the list of images for a composite image file in the database of composite images, determining whether any of the image files in the list of image files has been updated since the composite image file associated with the first web page was generated, and

in accordance with a determination that at least one image in the list of image files has been updated since the composite image file was generated, regenerating the composite image file.

9. The method of claim 8, wherein each image file and composite image file has an associated last-updated date, and wherein determining whether any of the image files in the list of image files has been updated since the composite image file was generated includes:

determining a last-updated date for the composite image file;

determining whether at least one image file in the list of image files has the last-updated date after the last-updated date for the composite image file; and

in accordance with the determination that at least one of the image files in the list of image files has the last-updated date after the last-updated date for

the composite image file, determining that at least one image file in the list of image files has been updated since the composite image file was generated.

10. The method of claim 1, wherein identifying a list of one or more images referenced in the requested first webpage further includes:

transmitting the webpage to the first client system, and

receiving a list of images requested by the first webpage from the first client system.

11. A server system comprising:

a receiving module to receive a request for a first webpage from a first client system

an identifying module to identify a list of one or more images referenced in the requested first webpage;

a determination module for determining whether the identified list of one or more images referenced in the requested first webpage matches a list of images for a specific composite image file in the database of composite images;

a retrieval module to, in accordance with a determination that the identified list of one or more images referenced in the requested first webpage matches a list of images for a specific composite image file in the database of composite images, retrieve the specific composite image and transmit the specific composite image to the first client system; and

a generation module to, in accordance with a determination that the identified list of one or more images referenced in the requested first webpage does match a list of images for a specific composite image file in the database of composite images, generating a composite image file from the identified list of one or more images and transmit the generated composite image file to the first client system.

12. The system of claim 11, wherein the first webpage is dynamically generated by the server system.

13. The system of claim 11, wherein the request for the first webpage includes a search query.

14. The system of claim 13, further comprising a storage module to store the one or more composite images in a database of composite images, wherein each composite image includes a list of images that are included in the composite image.

15. The system of claim 13, wherein the module to identify a list of one or more images referenced in the requested first webpage further includes:

a transmission module to transmit the webpage to the first client system, and

a list reception module to receive a list of images requested by the first webpage from the first client system.

16. A non-transitory computer-readable storage medium storing instructions that, when executed by the one or more processors of a machine, cause the machine to perform operations comprising:

receiving a request for a first webpage from a first client system

identifying a list of one or more images referenced in the requested first webpage;

compiling the identified one or more images into one or more first files;

transmitting the one or more first files to the first client system.

17. The non-transitory computer-readable storage medium of claim 16, wherein the first webpage is dynamically generated by the server system.

18. The non-transitory computer-readable storage medium of claim 16, wherein the request for the first webpage includes a search query.

19. The non-transitory computer-readable storage medium of claim 16, further comprising storing the one or more composite images in a database of composite images, wherein each composite image includes a list of images that are included in the composite image.

20. The non-transitory computer-readable storage medium of claim 19, wherein prior to compiling the identified one or more images into one or more first files, determining whether the identified list of one or more images matches the list of images for a composite image file in the database of composite images.

21. A computer readable medium carrying instructions that, when executed by the one or more processors of a machine, cause the machine to carry out the method of any one of claims 1 to 10.

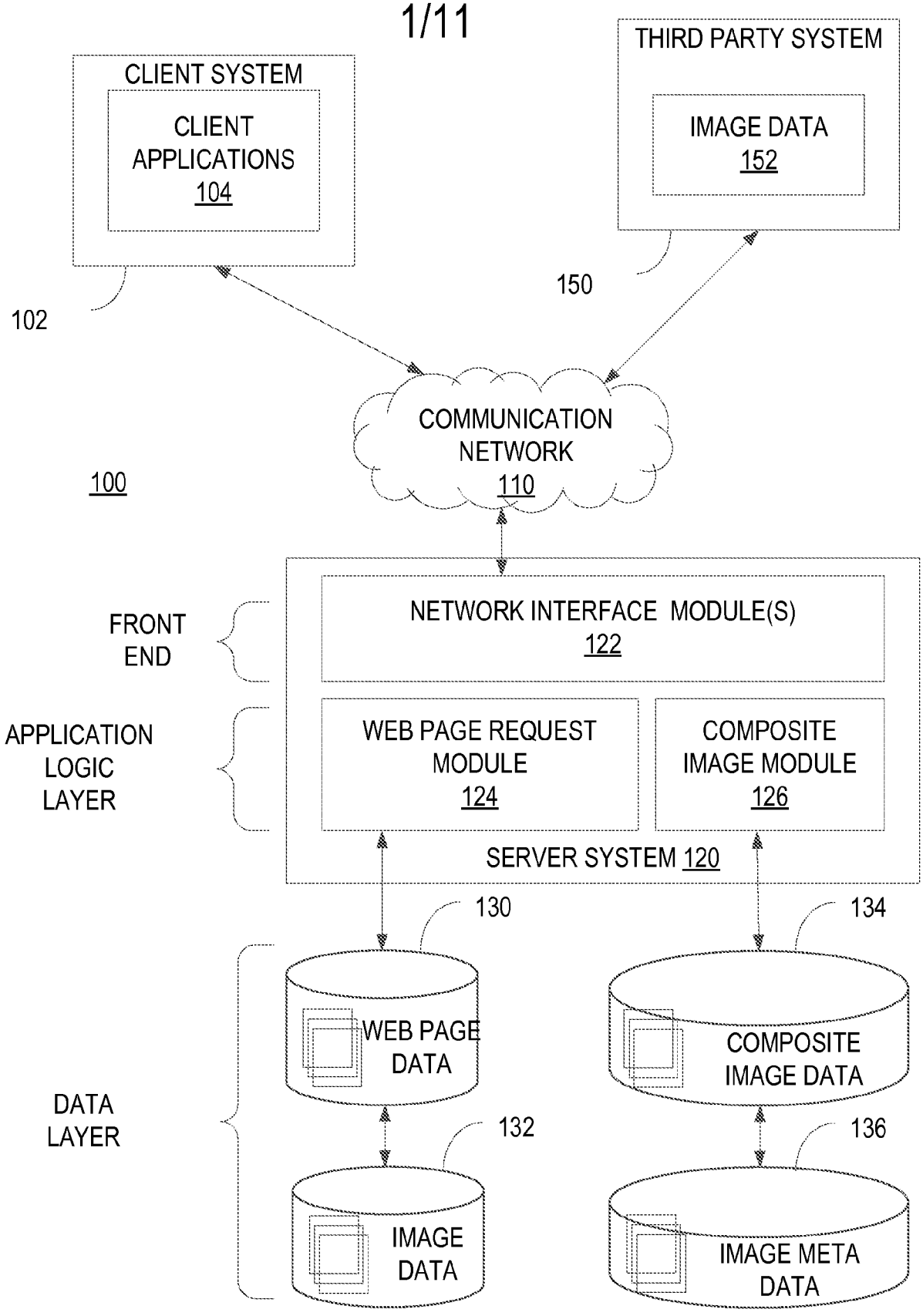
CLIENT SYSTEM
CLIENT APPLICATIONS 104
102
THIRD PARTY SYSTEM
IMAGE DATA 152
150
COMMUNICATION NETWORK 110
100
FRONT END
NETWORK INTERFACE MODULE(S) 122
APPLICATION LOGIC LAYER
WEB PAGE REQUEST MODULE 124
COMPOSITE IMAGE MODULE 126
SERVER SYSTEM 120
130
134
WEB PAGE DATA
COMPOSITE IMAGE DATA
DATA LAYER
132
136
IMAGE DATA
IMAGE META DATA

FIG. 1

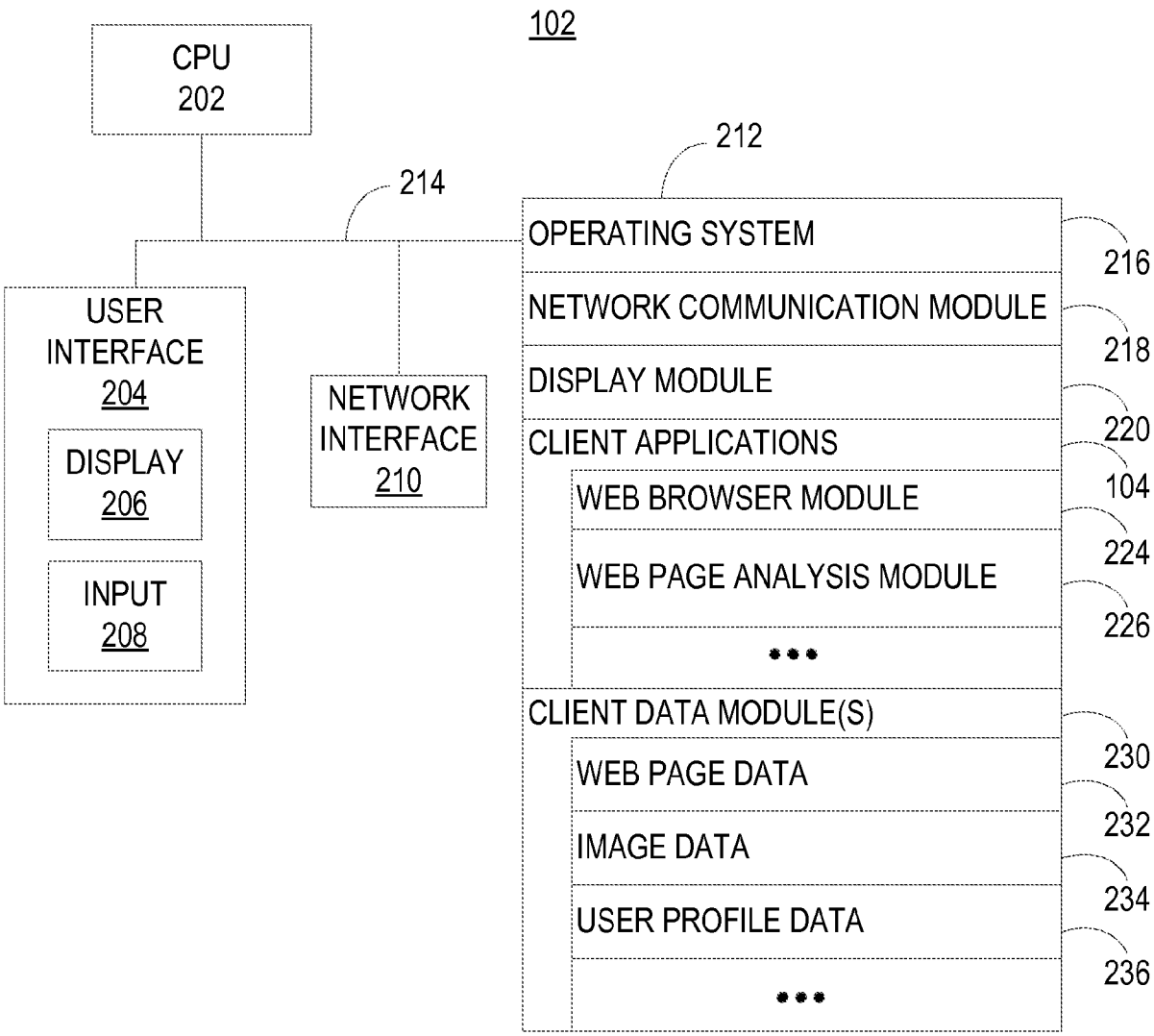
102
CPU
202
212
214
OPERATING SYSTEM
216
NETWORK COMMUNICATION MODULE
218
DISPLAY MODULE
220
CLIENT APPLICATIONS
104
WEB BROWSER MODULE
224
WEB PAGE ANALYSIS MODULE
226
CLIENT DATA MODULE(S)
230
WEB PAGE DATA
232
IMAGE DATA
234
USER PROFILE DATA
236
USER INTERFACE
204
DISPLAY
206
INPUT
208
NETWORK INTERFACE
210

FIG. 2

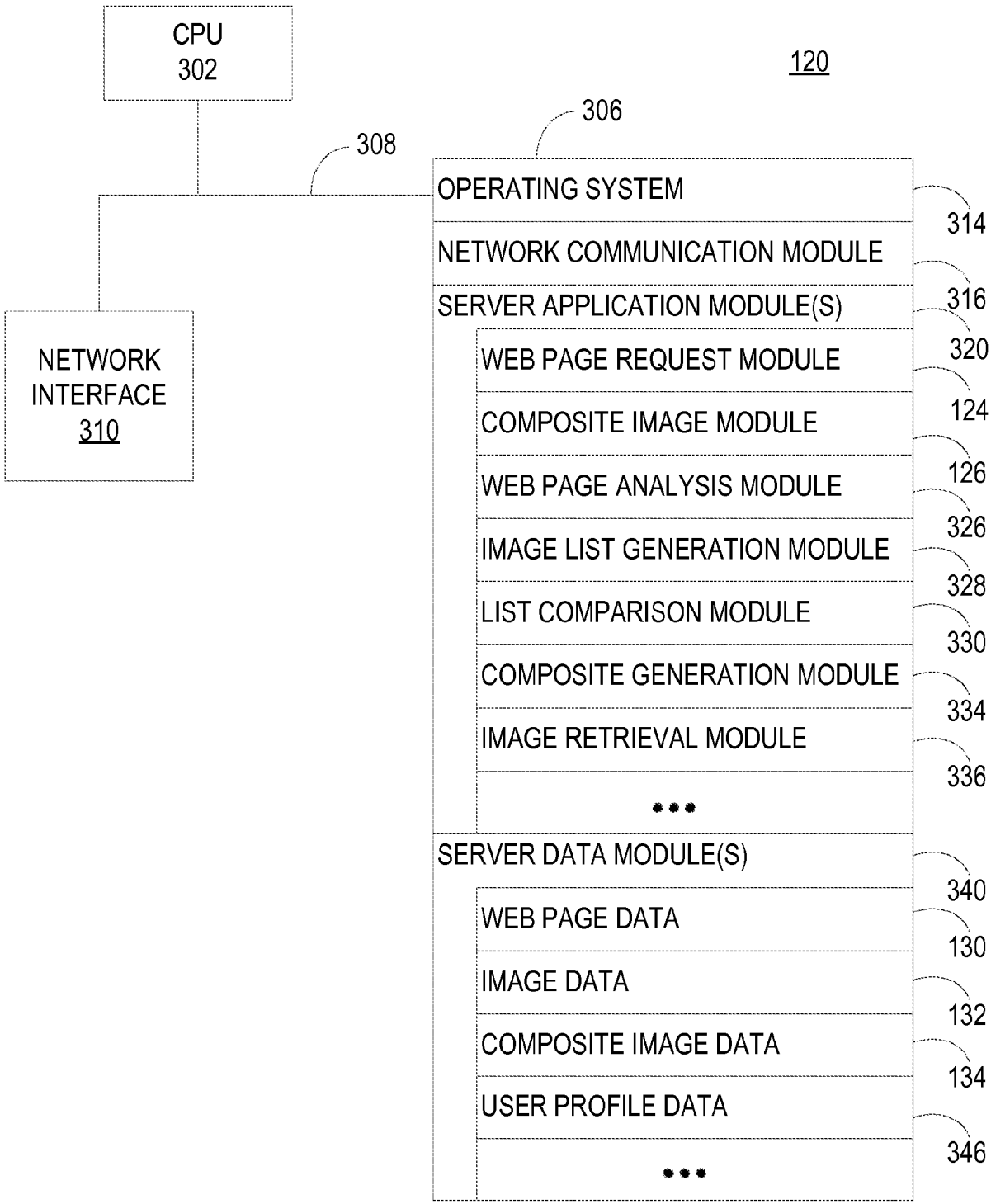
CPU
302
120
306
308
OPERATING SYSTEM
314
NETWORK COMMUNICATION MODULE
316
SERVER APPLICATION MODULE(S)
320
WEB PAGE REQUEST MODULE
124
COMPOSITE IMAGE MODULE
126
WEB PAGE ANALYSIS MODULE
326
IMAGE LIST GENERATION MODULE
328
LIST COMPARISON MODULE
330
COMPOSITE GENERATION MODULE
334
IMAGE RETRIEVAL MODULE
336
SERVER DATA MODULE(S)
340
WEB PAGE DATA
130
IMAGE DATA
132
COMPOSITE IMAGE DATA
134
USER PROFILE DATA
346
NETWORK
INTERFACE
310

FIG. 3

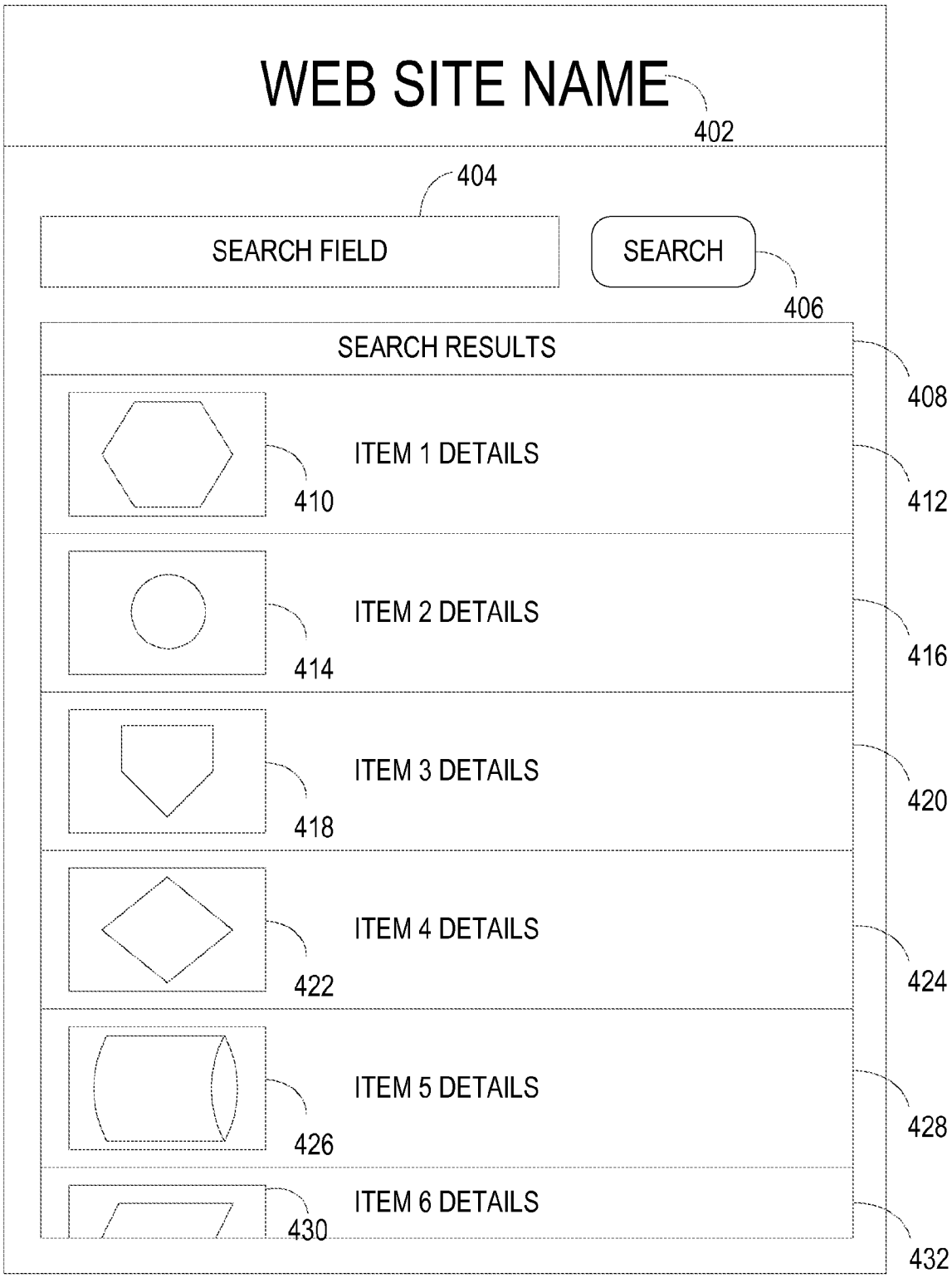
WEB SITE NAME
402
404
SEARCH FIELD
SEARCH
406
SEARCH RESULTS
408
ITEM 1 DETAILS
410
412
ITEM 2 DETAILS
414
416
ITEM 3 DETAILS
418
420
ITEM 4 DETAILS
422
424
ITEM 5 DETAILS
426
428
ITEM 6 DETAILS
430
432

FIG. 4A 400

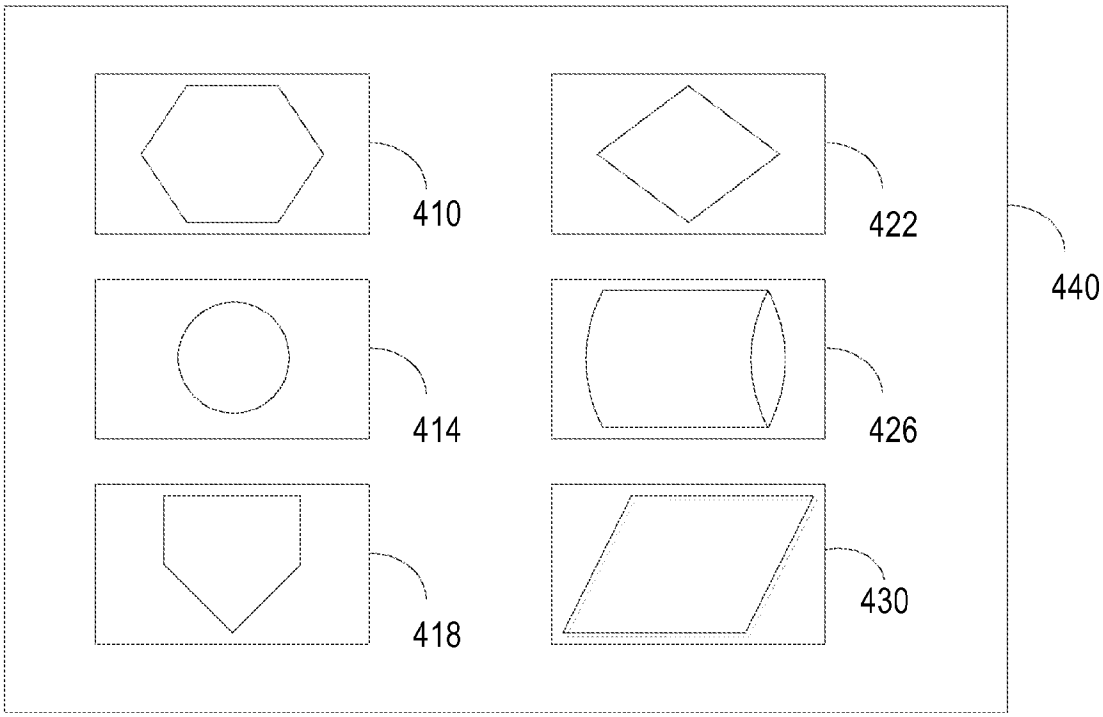

FIG. 4B

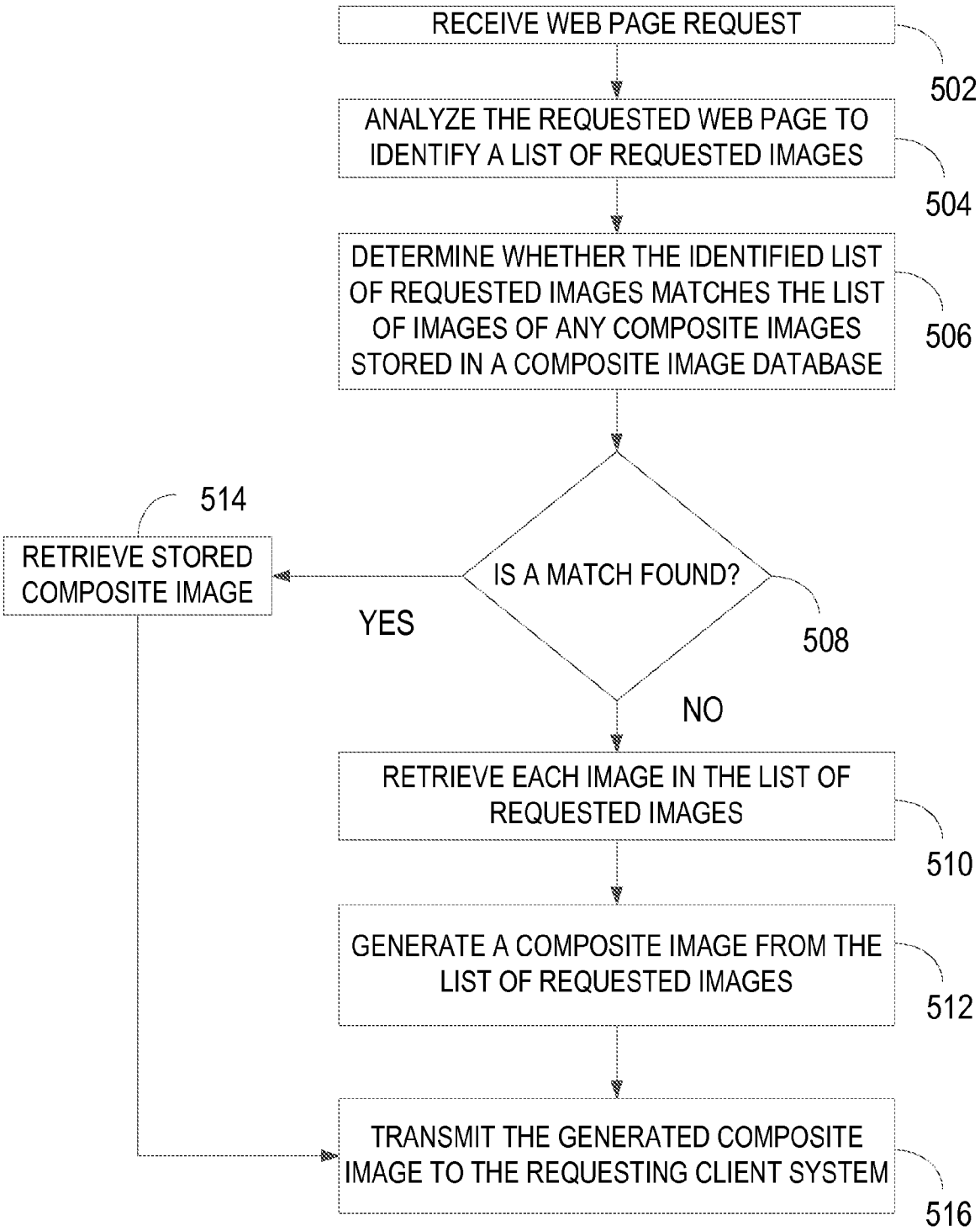
RECEIVE WEB PAGE REQUEST
502
ANALYZE THE REQUESTED WEB PAGE TO IDENTIFY A LIST OF REQUESTED IMAGES
504
DETERMINE WHETHER THE IDENTIFIED LIST OF REQUESTED IMAGES MATCHES THE LIST OF IMAGES OF ANY COMPOSITE IMAGES STORED IN A COMPOSITE IMAGE DATABASE
506
IS A MATCH FOUND?
508
YES
514
RETRIEVE STORED COMPOSITE IMAGE
NO
RETRIEVE EACH IMAGE IN THE LIST OF REQUESTED IMAGES
510
GENERATE A COMPOSITE IMAGE FROM THE LIST OF REQUESTED IMAGES
512
TRANSMIT THE GENERATED COMPOSITE IMAGE TO THE REQUESTING CLIENT SYSTEM
516

FIG. 5

500

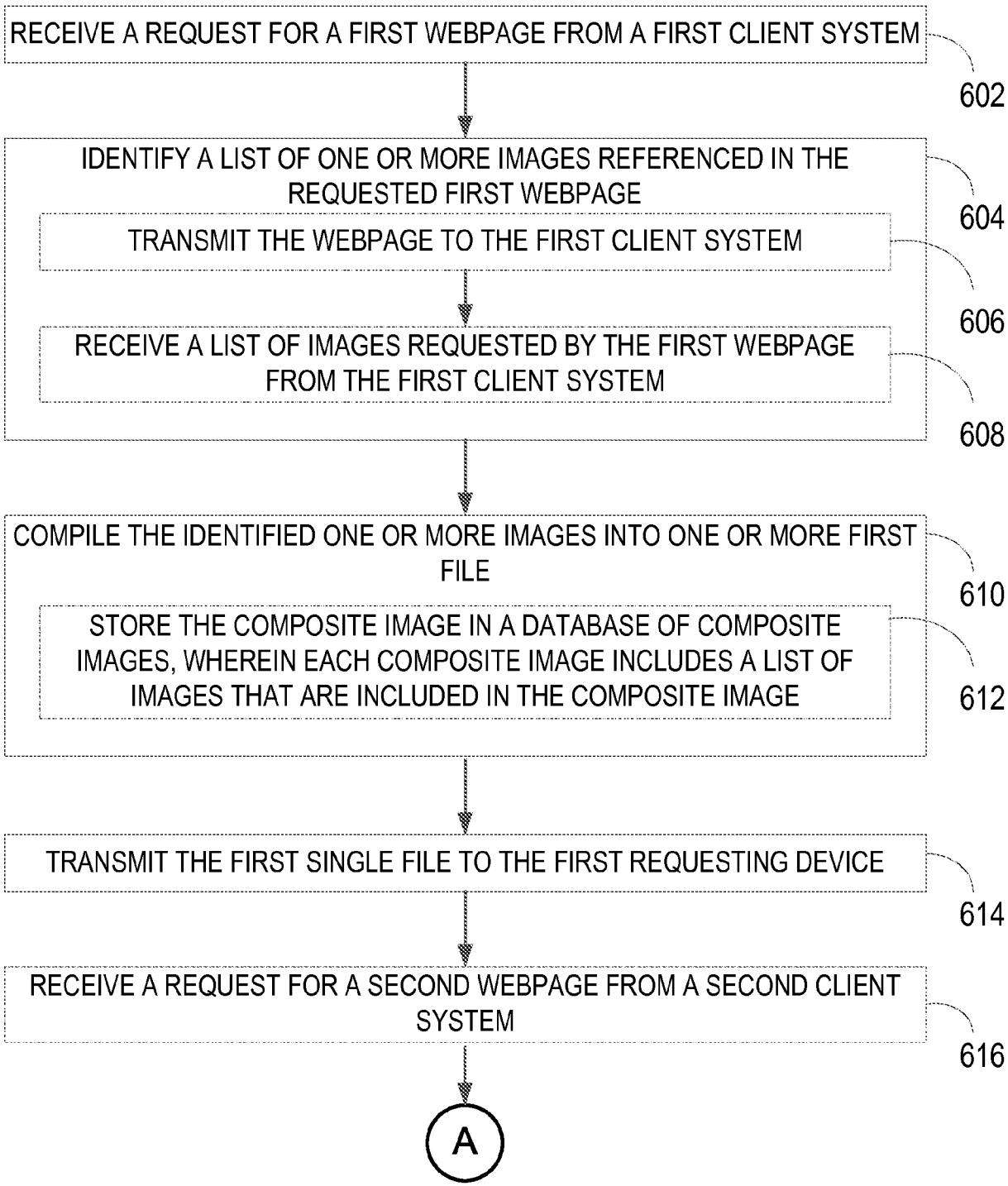
RECEIVE A REQUEST FOR A FIRST WEBPAGE FROM A FIRST CLIENT SYSTEM
602
IDENTIFY A LIST OF ONE OR MORE IMAGES REFERENCED IN THE REQUESTED FIRST WEBPAGE
604
TRANSMIT THE WEBPAGE TO THE FIRST CLIENT SYSTEM
606
RECEIVE A LIST OF IMAGES REQUESTED BY THE FIRST WEBPAGE FROM THE FIRST CLIENT SYSTEM
608
COMPILE THE IDENTIFIED ONE OR MORE IMAGES INTO ONE OR MORE FIRST FILE
610
STORE THE COMPOSITE IMAGE IN A DATABASE OF COMPOSITE IMAGES, WHEREIN EACH COMPOSITE IMAGE INCLUDES A LIST OF IMAGES THAT ARE INCLUDED IN THE COMPOSITE IMAGE
612
TRANSMIT THE FIRST SINGLE FILE TO THE FIRST REQUESTING DEVICE
614
RECEIVE A REQUEST FOR A SECOND WEBPAGE FROM A SECOND CLIENT SYSTEM
616
A

FIG. 6A

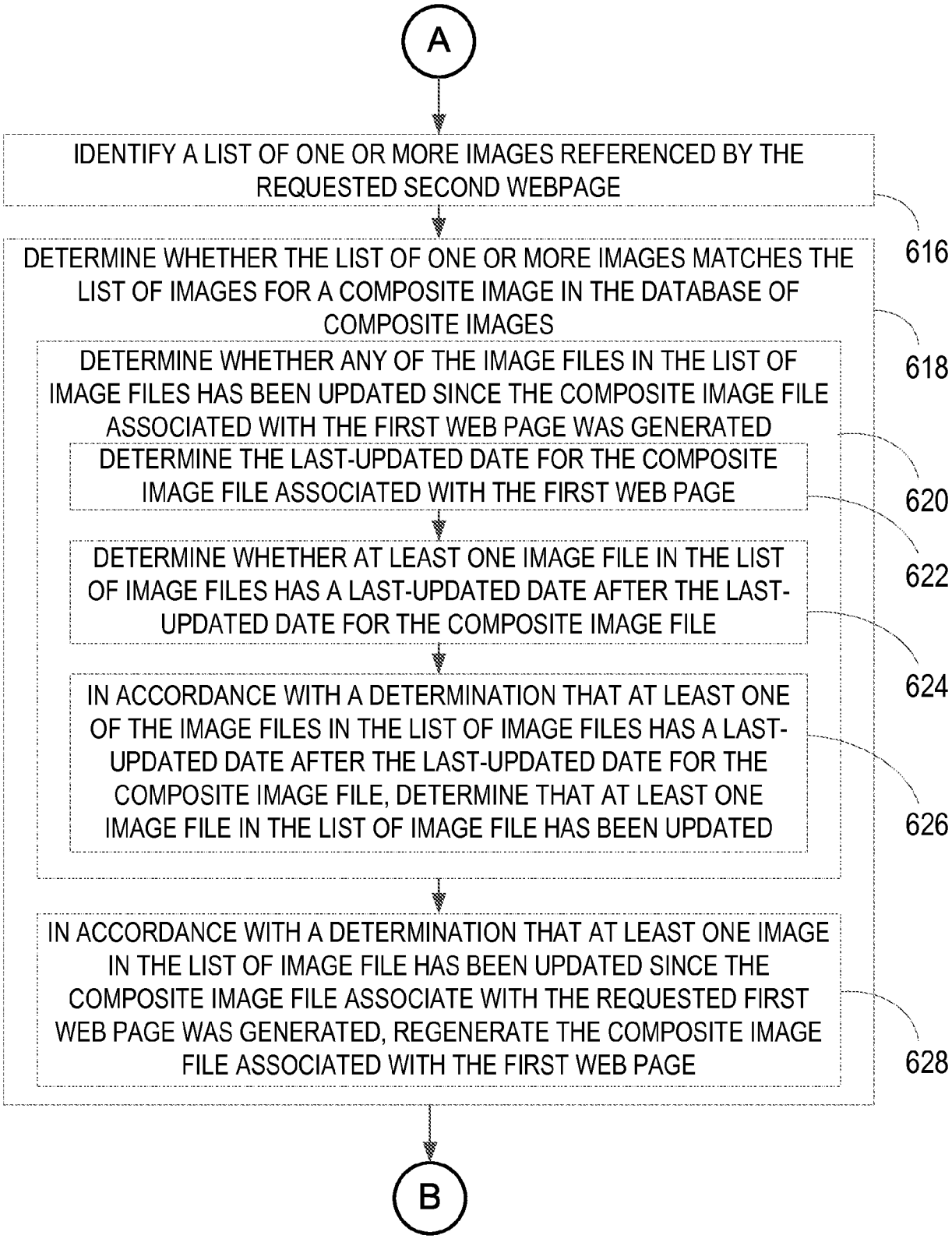
A
IDENTIFY A LIST OF ONE OR MORE IMAGES REFERENCED BY THE REQUESTED SECOND WEBPAGE
616
DETERMINE WHETHER THE LIST OF ONE OR MORE IMAGES MATCHES THE LIST OF IMAGES FOR A COMPOSITE IMAGE IN THE DATABASE OF COMPOSITE IMAGES
618
DETERMINE WHETHER ANY OF THE IMAGE FILES IN THE LIST OF IMAGE FILES HAS BEEN UPDATED SINCE THE COMPOSITE IMAGE FILE ASSOCIATED WITH THE FIRST WEB PAGE WAS GENERATED
DETERMINE THE LAST-UPDATED DATE FOR THE COMPOSITE IMAGE FILE ASSOCIATED WITH THE FIRST WEB PAGE
620
DETERMINE WHETHER AT LEAST ONE IMAGE FILE IN THE LIST OF IMAGE FILES HAS A LAST-UPDATED DATE AFTER THE LAST-UPDATED DATE FOR THE COMPOSITE IMAGE FILE
622
IN ACCORDANCE WITH A DETERMINATION THAT AT LEAST ONE OF THE IMAGE FILES IN THE LIST OF IMAGE FILES HAS A LAST-UPDATED DATE AFTER THE LAST-UPDATED DATE FOR THE COMPOSITE IMAGE FILE, DETERMINE THAT AT LEAST ONE IMAGE FILE IN THE LIST OF IMAGE FILE HAS BEEN UPDATED
624
626
IN ACCORDANCE WITH A DETERMINATION THAT AT LEAST ONE IMAGE IN THE LIST OF IMAGE FILE HAS BEEN UPDATED SINCE THE COMPOSITE IMAGE FILE ASSOCIATE WITH THE REQUESTED FIRST WEB PAGE WAS GENERATED, REGENERATE THE COMPOSITE IMAGE FILE ASSOCIATED WITH THE FIRST WEB PAGE
628
B

FIG. 6B

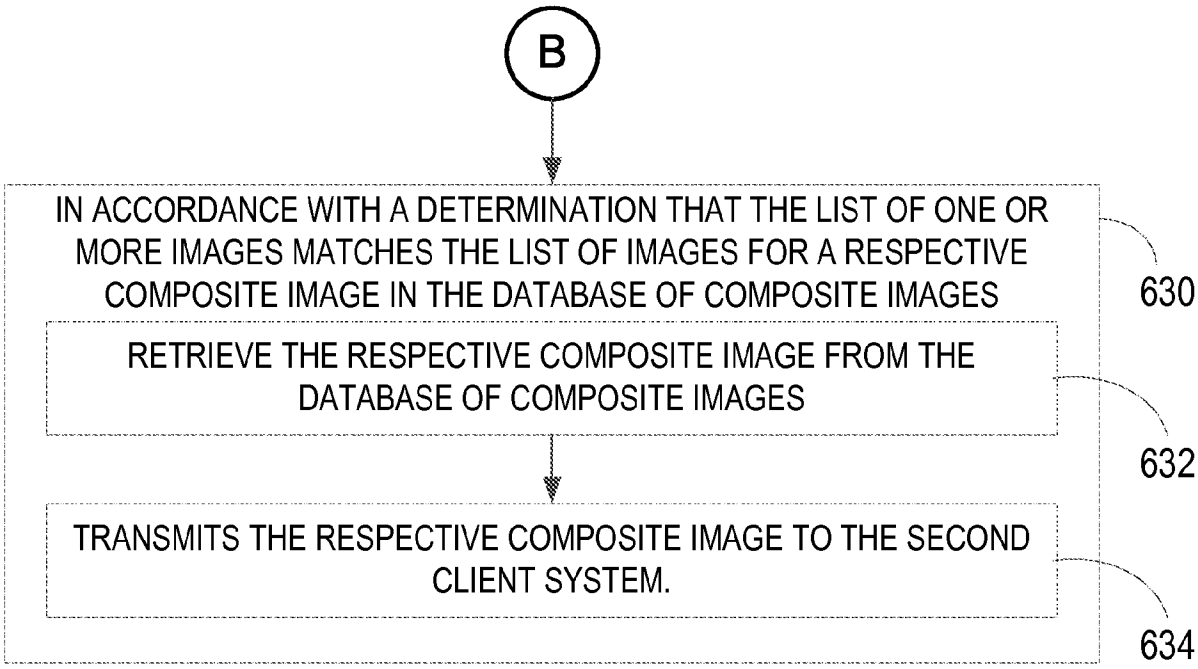
B
IN ACCORDANCE WITH A DETERMINATION THAT THE LIST OF ONE OR MORE IMAGES MATCHES THE LIST OF IMAGES FOR A RESPECTIVE COMPOSITE IMAGE IN THE DATABASE OF COMPOSITE IMAGES
630
RETRIEVE THE RESPECTIVE COMPOSITE IMAGE FROM THE DATABASE OF COMPOSITE IMAGES
632
TRANSMITS THE RESPECTIVE COMPOSITE IMAGE TO THE SECOND CLIENT SYSTEM.
634

FIG. 6C

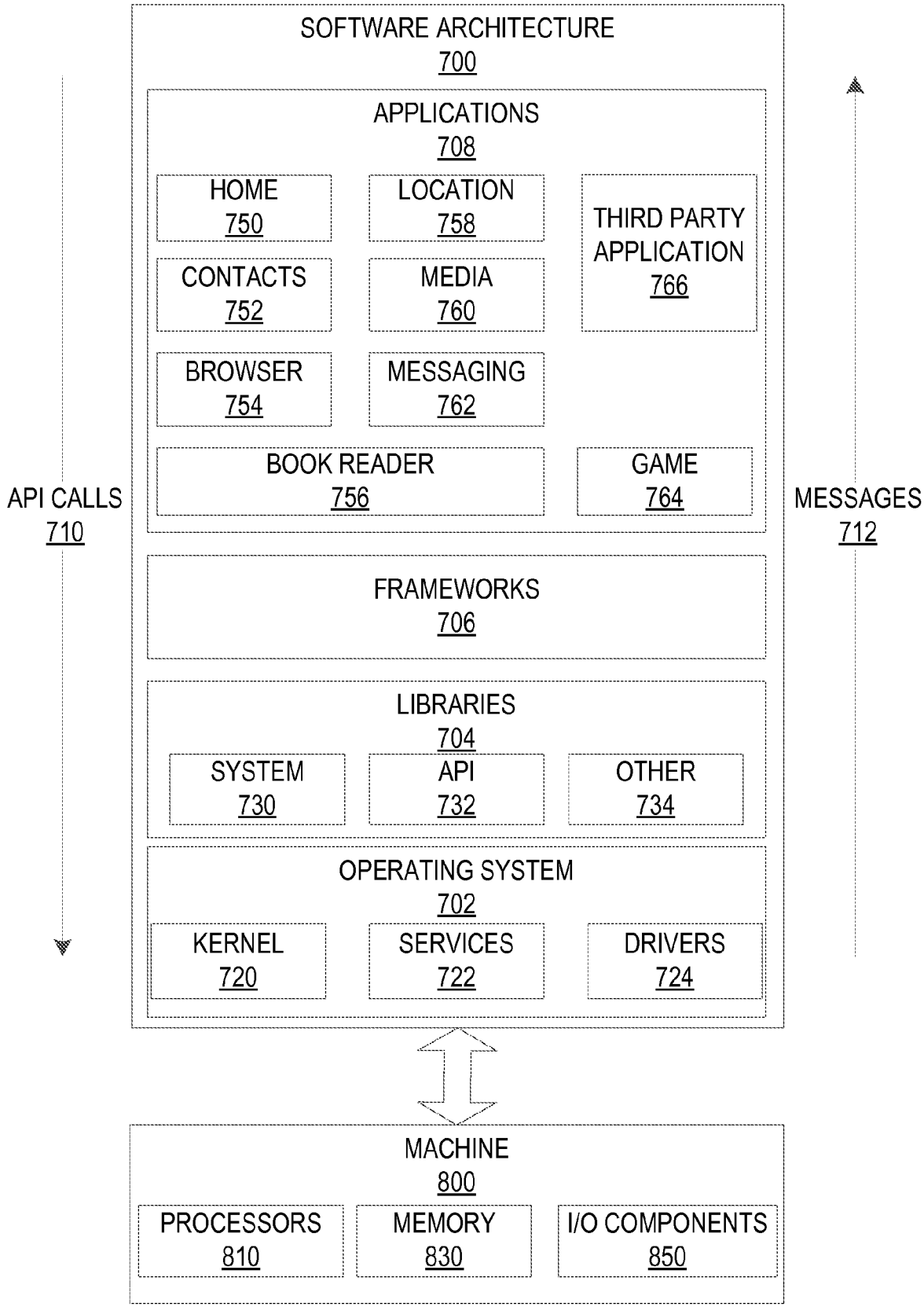
SOFTWARE ARCHITECTURE
700
APPLICATIONS
708
HOME
750
LOCATION
758
THIRD PARTY APPLICATION
766
CONTACTS
752
MEDIA
760
BROWSER
754
MESSAGING
762
BOOK READER
756
GAME
764
API CALLS
710
MESSAGES
712
FRAMEWORKS
706
LIBRARIES
704
SYSTEM
730
API
732
OTHER
734
OPERATING SYSTEM
702
KERNEL
720
SERVICES
722
DRIVERS
724
MACHINE
800
PROCESSORS
810
MEMORY
830
I/O COMPONENTS
850

FIG. 7

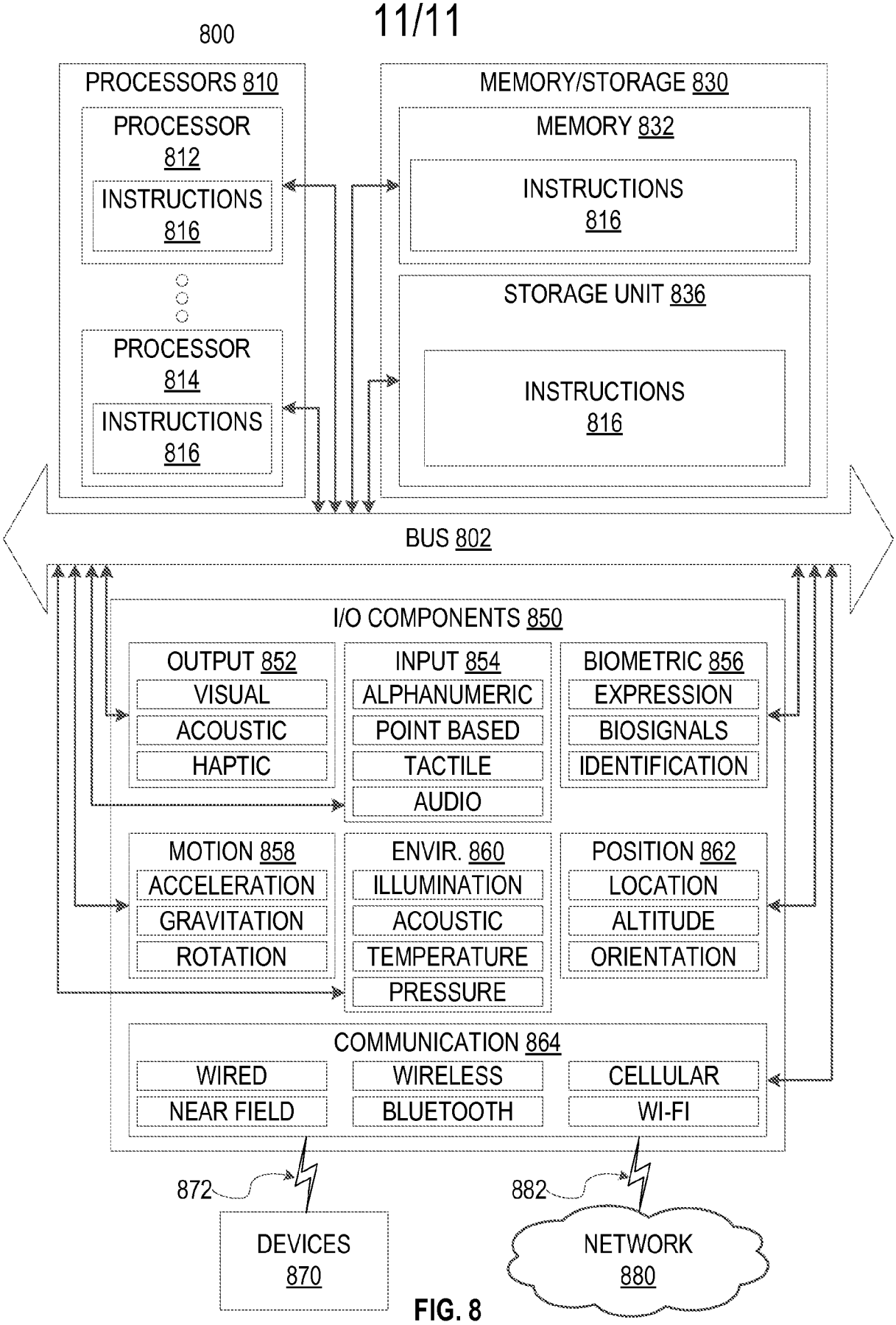
800
PROCESSORS 810
PROCESSOR 812
INSTRUCTIONS 816
PROCESSOR 814
INSTRUCTIONS 816
MEMORY/STORAGE 830
MEMORY 832
INSTRUCTIONS 816
STORAGE UNIT 836
INSTRUCTIONS 816
BUS 802
I/O COMPONENTS 850
OUTPUT 852
VISUAL
ACOUSTIC
HAPTIC
INPUT 854
ALPHANUMERIC
POINT BASED
TACTILE
AUDIO
BIOMETRIC 856
EXPRESSION
BIOSIGNALS
IDENTIFICATION
MOTION 858
ACCELERATION
GRAVITATION
ROTATION
ENVIR. 860
ILLUMINATION
ACOUSTIC
TEMPERATURE
PRESSURE
POSITION 862
LOCATION
ALTITUDE
ORIENTATION
COMMUNICATION 864
WIRED
NEAR FIELD
WIRELESS
BLUETOOTH
CELLULAR
WI-FI
872
DEVICES 870
882
NETWORK 880

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 15/56943

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 7/00 (2015.01)
CPC - G06F 17/30864
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)
USPC: 707/705, 707/736, 707/737; IPC(8): G06F 7/00 (2015.01);
CPC: G06F17/30864, G06Q10/10, G06Q30/02, G06F17/30867, G06F17/30011

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 382/284, 345/629, 358/450, 709/203; IPC(8): H04N1/387, H04N1/23, G06T3/00, G03G15/36, G06K9/36 (2015.01)
CPC: H04N1/00185, H04N1/3876, G06T11/60

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PatBase, Google Patents, Google Web
Keywords: website, image, server, database, composite, merge, combine, update, alter, change, request, generate

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X ---- Y	US 2011/0239108 A1 (Blomquist et al.) 29 September 2011 (29.09.2011), entire document, especially abstract, para [0019], [0021]-[0026], [0037], [0042].	1-5, 7-12, 16-17, 19-20, 21/(1-5, 7-10) ---------- 6, 13-15, 18, 21/6
Y	US 8,010,624 B2 (Scott et al.) 30 August 2011 (30.08.2011), abstract, col 3, ln 22-44.	6, 13-15, 18, 21/6

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

* 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
04 December 2015	19 JAN 2016
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (January 2015)