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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
06 August 2020 (06.08.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/159898 A1

(51) International Patent Classification:
G06F 21/33 (2013.01) *H04L 9/32* (2006.01)
G06T 1/00 (2006.01)

(21) International Application Number:
PCT/US2020/015283

(22) International Filing Date:
28 January 2020 (28.01.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
16/263,667 31 January 2019 (31.01.2019) US

(71) Applicant: **MCAFEE, LLC** [US/US]; 2821 Mission College Boulevard, Santa Clara, California 95054-1838 (US).

(72) Inventors: **ZHENG, Yi**; 106 Lancewood Pl., Los Gatos, California 95032 (US). **SANZGIRI, Ameya M.**; 37734 Elliot Street, Fremont, California 94536 (US).

(74) Agent: **PEMBERTON, John D.**; Patent Capital Group, 2816 Lago Vista Lane, Rockwall, Texas 75032 (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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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).

Declarations under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

(54) Title: SYSTEM AND METHOD FOR APPLICATION EXPLORATION

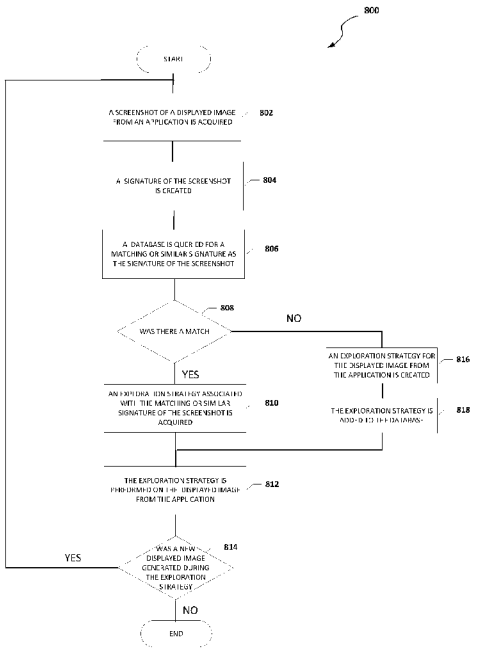


FIGURE 8

(57) Abstract: Particular embodiments described herein provide for an electronic device that can be configured to capture an image on a display, where the image includes at least one user interface element and is part of an application, create a screen signature of the image, determine an exploration strategy for the image based on the screen signature, and perform the exploration strategy on the image. The image can be abstracted to create the screen signature and the exploration strategy includes interacting with each of the at least one user interface elements.

WO 2020/159898 A1

[Continued on next page]

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))*

SYSTEM AND METHOD FOR APPLICATION EXPLORATION

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of and priority from U.S. Patent Application Serial No. 16/263,667, filed January 31, 2019, entitled "SYSTEM AND METHOD FOR APPLICATION EXPLORATION", which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] This disclosure relates in general to the field of information security, and more particularly, to a system and method for application exploration.

BACKGROUND

[0003] The field of network security has become increasingly important in today's society. The Internet has enabled interconnection of different computer networks all over the world. In particular, the Internet provides a medium for exchanging data between different users connected to different computer networks via various types of client devices. While the use of the Internet has transformed business and personal communications, it has also been used as a vehicle for malicious operators to gain unauthorized access to computers and computer networks and for intentional or inadvertent disclosure of sensitive information.

[0004] Malicious software ("malware") that infects a host computer may be able to perform any number of malicious actions, such as stealing sensitive information from a business or individual associated with the host computer, propagating to other host computers, and/or assisting with distributed denial of service attacks, sending out spam or malicious emails from the host computer, etc. Hence, administrative challenges remain for protecting computers and computer networks from malicious and inadvertent exploitation by malicious software and/or malicious operators.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] To provide a better understanding of the present disclosure and features and advantages thereof, reference is made to the following description, taken in conjunction with the accompanying figures, wherein like reference numerals represent like parts, in which:

[0006] FIGURE 1 is a simplified block diagram of a system to help facilitate a system and method for application exploration in accordance with an embodiment of the present disclosure;

[0007] FIGURE 2 is a simplified block diagram of a portion of a system to help facilitate a system and method for application exploration in accordance with an embodiment of the present disclosure;

[0008] FIGURE 3 is a simplified block diagram of a portion of a system illustrating example details to help facilitate a system and method for application exploration in accordance with an embodiment of the present disclosure;

[0009] FIGURES 4A and 4B are a simplified block diagram of a portion of a system illustrating example details to help facilitate a system and method for application exploration in accordance with an embodiment of the present disclosure;

[0010] FIGURE 5 is a simplified block diagram of a portion of a system illustrating example details to help facilitate a system and method for application exploration in accordance with an embodiment of the present disclosure;

[0011] FIGURE 6 is a simplified flowchart illustrating potential operations that may be associated with the system in accordance with an embodiment;

[0012] FIGURE 7 is a simplified flowchart illustrating potential operations that may be associated with the system in accordance with an embodiment;

[0013] FIGURE 8 is a simplified flowchart illustrating potential operations that may be associated with the system in accordance with an embodiment;

[0014] FIGURE 9 is a simplified flowchart illustrating potential operations that may be associated with the system in accordance with an embodiment;

[0015] FIGURE 10 is a simplified flowchart illustrating potential operations that may be associated with the system in accordance with an embodiment;

[0016] FIGURE 11 is a block diagram illustrating an example computing system that is arranged in a point-to-point configuration in accordance with an embodiment;

[0017] FIGURE 12 is a simplified block diagram associated with an example advanced reduced instruction set computer machine (ARM) ecosystem system on chip (SOC) of the present disclosure; and

[0018] FIGURE 13 is a block diagram illustrating an example processor core in accordance with an embodiment.

[0019] The FIGURES of the drawings are not necessarily drawn to scale, as their dimensions can be varied considerably without departing from the scope of the present disclosure.

DETAILED DESCRIPTION OF EMBODIMENTS

[0020] The following detailed description sets forth examples of apparatuses, methods, and systems relating to an application exploration in accordance with an embodiment of the present disclosure. Features such as structure(s), function(s), and/or characteristic(s), for example, are described with reference to one embodiment as a matter of convenience; various embodiments may be implemented with any suitable one or more of the described features.

[0021] In the following description, various aspects of the illustrative implementations will be described using terms commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. However, it will be apparent to those skilled in the art that the embodiments disclosed herein may be practiced with only some of the described aspects. For purposes of explanation, specific numbers, materials, and configurations are set forth in order to provide a thorough understanding of the illustrative implementations. However, it will be apparent to one skilled in the art that embodiments such as disclosed herein may be practiced without these specific details. In other instances, known features are omitted or simplified in order not to obscure the illustrative implementations.

[0022] In the following detailed description, reference is made to the accompanying drawings that form a part hereof wherein like numerals designate like parts throughout, and in which is shown, by way of illustration, embodiments that may be practiced. It is to be

understood that other embodiments may be utilized and structural or logical changes may be made without departing from the scope of the present disclosure. Therefore, the following detailed description is not to be taken in a limiting sense. For the purposes of the present disclosure, the phrase "A and/or B" means (A), (B), or (A and B). For the purposes of the present disclosure, the phrase "A, B, and/or C" means (A), (B), (C), (A and B), (A and C), (B and C), or (A, B, and C).

[0023] FIGURE 1 is a simplified block diagram of a system 100 to help facilitate an application exploration in accordance with an embodiment of the present disclosure. As illustrated in FIGURE 1, an embodiment of system 100 can include network elements 102a–102c, a server 104, and cloud services 106. Network elements 102a-102c, server 104, and cloud services 106 may be in communication with each other using network 108.

[0024] Each of network elements 102a–102c, server 104, and/or cloud services 106 can include memory, a computer processing unit (CPU), a security engine, one or more applications, and a display. For example, as illustrated in FIGURE 1, network element 102a, server 104, and cloud services 106 include memory 110, one or more CPUs 112, a security engine 114, one or more applications 116a-116c, and a display 122. The memory can include one or more files and a screen signature database. For example, as illustrated in FIGURE 1, memory 110 can include one or more files 118 and a screen signature database 120.

[0025] Display 122 may be a physical display or a virtual or emulated display. The phases “displayed on a display,” “displayed image,” and derivative phrases is to include being displayed on a physical display and/or a virtual or emulated display. One or more of applications 116a-116c may be used by a malicious operator, may include exploits that can be used by a malicious operator, may expose sensitive data, may be a poorly designed application, etc. For example, malware 124 in network element 102b may use an exploit in application 116c to conduct malicious activity or may cause a malicious file (e.g., file 118) to be installed in network element 102a.

[0026] To help determine if an application (e.g., application 116a) may be used by a malicious operator, may include exploits that can be used by a malicious operator, may expose sensitive data, may be a poorly designed application, etc., an application analysis can be performed on the application. An application analysis involves analyzing the application and exploring and interacting with the application to try and determine what operating

system calls are made, what kind of data being written, what data is being transferred, what types of data are being accessed, what input is being requested, etc. The results of the application analysis can be analyzed to determine if there is some type of malicious activity, if there are privacy issues, the application has user interface issues, the application is benign, etc.

[0027] Security engine 114 can be configured to explore and interact with an application and perform an application analysis. For purposes of this disclosure, security engine 114 is not configured to detect malware, determine if the application includes exploits, if the application exposes sensitive data or includes privacy issues, etc. but those features can be included in security engine 114 if desired. For example, to detect malware behavior from ransomware, a feature can be added to record the images on a display that have been explored and match the explored images with a ransomware database that includes strings or images that are indicative of ransomware behavior (e.g., an image or screen asking for bitcoins). In another example, to identify privacy data leaking, a feature can be added to record the traffic data that an application generates while the application is being explored and then compare the traffic data with a Uniform Resource Locator (URL) reputation database and determine if requests to malicious or suspicious URLs contain privacy information. Security engine 114 can be configured explore and interact with an application to determine what images an application causes to be displayed on a display that a user interacts with, how data is entered, how forms are filled out, what and how icons are clicked or selected, etc.

[0028] In an example, security engine 114 can be configured to capture or take a screenshot of a target image that is part of an application. More specifically, an application or file can cause the target image to be displayed on a display (e.g., display 122) for user interaction (e.g., the target image includes one or more user interface elements). The screenshot of the target image can be analyzed to create a screen signature. In an example, the screen signature can be an abstracted mathematical representation of the target image. The screen signature of the target image can be compared to known screen signatures in screen signature database 120.

[0029] Each of the screen signatures in screen signature database 120 are associated with an exploration strategy. The exploration strategy is a preset way of exploring an image that has a specific screen signature and the exploration strategy mimics a user's interaction

with the user interface elements in the image. An exploration strategy can include a series of actions, including tap, scroll, input character, etc. that can be performed by a user as they interact with the target image. More specifically, the exploration strategy can include a series or set of events such as a taping on a certain area of the screen that includes the image, scrolling the image to the left or the right, scrolling the image up or down, inputting data, inputting a sequence of letters, numbers, and/or symbols (e.g., my_dummy_email@dummy_server.com), waiting for five (5) seconds, etc.). In an example, the series of actions in an exploration strategy may be a series of standardized actions. More specifically, the series of standardized actions may be one or more of the standardized actions by Linus Torvalds (2015) Linux (4.1-rc8) {operating system}, retrieved from https://www.kernel.org/doc/html/v4.17/_sources/input/event-codes.rst.txt, herein incorporated by reference.

[0030] If the exploration strategy of the target image generates a new image that is displayed on the display, then a new screen signature of the new image can be generated, compared to known screen signatures in screen signature database 120 to determine a new exploration strategy for the new image, and the new image can be can be explored using the new exploration strategy. By using the exploration strategy associated with the screen signature in screen signature database 120 that is similar to the screen signature of the target image and an exploration strategy associated with any images generated during the exploration of the target image, the application associated with the target image can be explored. This allows the system to perform an application analysis on an application and can help the system determine if an application may be used by a malicious operator, may include exploits that can be used by a malicious operator, may expose sensitive data, may be a poorly designed application, etc.

[0031] If two images are similar, then the same exploration strategy can be used to interact with the images instead of having to use a random or brute force approach towards choosing the user interface elements to interact with or asking a human to interact with the image and/or record the process. Security engine 114 can be configured to determine the similarity of the images so that an administrator does not need to do brute force searching of the image or for similar images. This is a scalable system for a large number of applications, can reduce the search space dramatically, and can allow for analysis and exploration of an

application without the need for an administrator to physically explore the application. The data in screen signature database 120 can be organized similar to a knowledge graph that can be searched using a Monte Carlo Tree Search or some other means that may be used to search screen signature database 120. Once an application has been explored, a separate process can analyze the results of the application exploration and attempt to determine if there is some type of malicious activity associated with the application, there are privacy issues associated with the application, the application has user interface issues, the application is benign, etc.

[0032] As used herein, the term "when" may be used to indicate the temporal nature of an event. For example, the phrase "event 'A' occurs when event 'B' occurs" is to be interpreted to mean that event A may occur before, during, or after the occurrence of event B, but is nonetheless associated with the occurrence of event B. For example, event A occurs when event B occurs if event A occurs in response to the occurrence of event B or in response to a signal indicating that event B has occurred, is occurring, or will occur. Reference to "one embodiment" or "an embodiment" in the present disclosure means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment. The appearances of the phrase "in one embodiment" or "in an embodiment" are not necessarily all referring to the same embodiment.

[0033] Elements of FIGURE 1 may be coupled to one another through one or more interfaces employing any suitable connections (wired or wireless), which provide viable pathways for network (e.g., network 108) communications. Additionally, any one or more of these elements of FIGURE 1 may be combined or removed from the architecture based on particular configuration needs. System 100 may include a configuration capable of transmission control protocol/Internet protocol (TCP/IP) communications for the transmission or reception of packets in a network. System 100 may also operate in conjunction with a user datagram protocol/IP (UDP/IP) or any other suitable protocol where appropriate and based on particular needs.

[0034] For purposes of illustrating certain example techniques of system 100, it is useful to understand the communications that may be traversing the network environment. The following foundational information may be viewed as a basis from which the present disclosure may be properly explained.

[0035] Malicious software ("malware") that infects a host computer may be able to perform any number of malicious actions, such as stealing sensitive information from a business or individual associated with the host computer, propagating to other host computers, assisting with distributed denial of service attacks, sending out spam or malicious emails from the host computer, etc. Hence, administrative challenges remain for protecting computers and computer networks from malicious and inadvertent exploitation by malicious software and devices. One-way security systems try to mitigate malware is to analyze applications and try to determine their risk to exploitation from malware or a malicious operator.

[0036] One of the most difficult challenges in application analysis is the automation of application exploration. The term "automation of application exploration," "automatic application exploration," and other similar terms that use the term "automatic" or derivations of the term "automatic" mean exploring one or more applications without human intervention or with minimal human intervention. As a core technology in the security and privacy domain, security analysists need to analyze and understand the security implications of their interaction with an application's elements before convicting applications. Towards this end, a security analysist can spend a lot of time trying to interact with applications to evaluate their risk for security, privacy, performance, etc. Therefore, what is needed is a system and method for application exploration.

[0037] A system and method to help facilitate an application exploration, as outlined in FIGURE 1, can resolve these issues (and others). Using security engine 114, system 100 can be configured to use image similarity to determine exploration strategies for one or more images that are part of an application. More specifically, a screenshot of a target image that an application causes to be displayed on a display can be taken and the elements in the screenshot can then be estimated into abstract objects (e.g., geometric shapes such as rectangles). The abstract objects are then either assimilated or separated depending on the coordinates of the abstract objects (e.g., abstract objects that are very close to each other are assimilated, the ones that are far apart are kept separate) and a screen signature of the target image can be generated. For example, a similarity engine can extract a layout outline from images using geometric shapes. The screen signature of the abstracted image can be matched to an image with a similar screen signature that has an associated exploration strategy. The

system can use the exploration strategy to explore the target image that is displayed on the display. By abstracting an image that an application causes to be displayed on a display, creating a screen signature of the image, and determining an exploration strategy for the image by using an exploration strategy of an image with a similar screen signature, the system allows for automated exploration of an application.

[0038] More specifically, instead of depending on access to an application's internal structure or code and an administrator manually exploring a target image displayed on a display, the system can use a known exploration strategy for an image that is similar to the target image to explore the target image. The exploration strategy is an exploration strategy that mimics a user's interaction with the target image displayed on a display. By using a known exploration strategy for an image that is similar to the target image displayed on the display, the system can determine the strategy of exploration for the target image that can increase coverage while minimizing (or at least reducing) time spent exploring the target image. This allows for the possibility of automation as instrumentation or recompiling by an administrator can be removed from the application exploration process. This makes the system extremely versatile as the system can abstract out the elements of the image in to geometric shapes and the system can become operating system agnostic. Being operating system agnostic eliminates the need for deeper analysis (e.g., call graph analysis) which is a requirement for current systems and is computationally inefficient. In some examples, the exploration strategy can be directly recorded, for example, by using the Android 'adb' command. Direct access to '/dev/input#' device on an Android device or an emulator can help the system to create an exploration strategy that can be used in cross-platform applications. For example, what is learned on an Android based operating system can be applied to iOS based operating system.

[0039] In an illustrative example, the system can capture the target image (e.g., take a screenshot of a target image) or an emulator can capture the target image (e.g., generate a screenshot) from a virtual representation of the target image. A screen signature can be created for the captured image and associated with the target image. The screen signature of the target image is used to query the screen signature database. The response to the query from the screen signature database can be one or more exploration strategies where each exploration strategy corresponds to a screen signature in the screen signature database that

is similar to the screen signature of the target image. The system can execute the one or more exploration strategies returned from the screen similarity database to explore the target image. In an example, the system can categorize images into small sets such that all the images in a particular set are similar and can be explored with the same exploration strategy.

[0040] Turning to the infrastructure of FIGURE 1, system 100 in accordance with an example embodiment is shown. Generally, system 100 can be implemented in any type or topology of networks. Network 108 represents a series of points or nodes of interconnected communication paths for receiving and transmitting packets of information that propagate through system 100. Network 108 offers a communicative interface between nodes, and may be configured as any local area network (LAN), virtual local area network (VLAN), wide area network (WAN), wireless local area network (WLAN), metropolitan area network (MAN), Intranet, Extranet, virtual private network (VPN), and any other appropriate architecture or system that facilitates communications in a network environment, or any suitable combination thereof, including wired and/or wireless communication.

[0041] In system 100, network traffic, which is inclusive of packets, frames, signals, data, etc., can be sent and received according to any suitable communication messaging protocols. Suitable communication messaging protocols can include a multi-layered scheme such as Open Systems Interconnection (OSI) model, or any derivations or variants thereof (e.g., Transmission Control Protocol/Internet Protocol (TCP/IP), user datagram protocol/IP (UDP/IP)). Additionally, radio signal communications over a cellular network may also be provided in system 100. Suitable interfaces and infrastructure may be provided to enable communication with the cellular network.

[0042] The term "packet" as used herein, refers to a unit of data that can be routed between a source node and a destination node on a packet switched network. A packet includes a source network address and a destination network address. These network addresses can be Internet Protocol (IP) addresses in a TCP/IP messaging protocol. The term "data" as used herein, refers to any type of binary, numeric, voice, video, textual, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another in electronic

devices and/or networks. Additionally, messages, requests, responses, and queries are forms of network traffic, and therefore, may comprise packets, frames, signals, data, etc.

[0043] Network elements 102a-102c can each be a desktop computer, laptop computer, mobile device, personal digital assistant, smartphone, tablet, network appliances, servers, routers, switches, gateways, bridges, load balancers, or any other suitable device, component, element, or object operable to exchange information in a network environment. Server 104 can be a network element such as a server or virtual server and can be associated with clients, customers, endpoints, or end users wishing to initiate a communication in system 100 via some network (e.g., network 108). The term 'server' is inclusive of devices used to serve the requests of clients and/or perform some computational task on behalf of clients within system 100. Cloud services 106 is configured to provide cloud services to network elements 102a-102c. Cloud services may generally be defined as the use of computing resources that are delivered as a service over a network, such as the Internet. Often, compute, storage, and network resources are offered in a cloud infrastructure, effectively shifting the workload from a local network to the cloud network. Network elements 102a-102c may include any suitable hardware, software, components, modules, or objects that facilitate the operations thereof, as well as suitable interfaces for receiving, transmitting, and/or otherwise communicating data or information in a network environment. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information.

[0044] In regards to the internal structure associated with system 100, each of network elements 102a-102c, server 104, and cloud services 106 can include memory elements (e.g., memory 110) for storing information to be used in the operations outlined herein. Each of network elements 102a-102c, server 104, and cloud services 106 may keep information in any suitable memory element (e.g., disk, random access memory (RAM), read-only memory (ROM), erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), application specific integrated circuit (ASIC), etc.), software, hardware, firmware, or in any other suitable component, device, element, or object where appropriate and based on particular needs. Any of the memory items discussed herein should be construed as being encompassed within the broad term 'memory element.' Moreover, the information being used, tracked, sent, or received in system 100 could be provided in any

database, register, queue, table, cache, control list, or other storage structure, all of which can be referenced at any suitable timeframe. Any such storage options may also be included within the broad term 'memory element' as used herein.

[0045] In certain example implementations, the functions outlined herein may be implemented by logic encoded in one or more tangible media (e.g., embedded logic provided in an ASIC, digital signal processor (DSP) instructions, software (potentially inclusive of object code and source code) to be executed by a processor, or other similar machine, etc.), which may be inclusive of non-transitory computer-readable media. In some of these instances, memory elements can store data used for the operations described herein. This includes the memory elements being able to store software, logic, code, or processor instructions that are executed to carry out the activities described herein.

[0046] In an example implementation, network elements of system 100, such as network elements 102a-102c, server 104, and cloud services 106 may include software modules (e.g., security engine 114) to achieve, or to foster, operations as outlined herein. These modules may be suitably combined in any appropriate manner, which may be based on particular configuration and/or provisioning needs. In example embodiments, such operations may be carried out by hardware, implemented externally to these elements, or included in some other network device to achieve the intended functionality. Furthermore, the modules can be implemented as software, hardware, firmware, or any suitable combination thereof. These elements may also include software (or reciprocating software) that can coordinate with other network elements in order to achieve the operations, as outlined herein.

[0047] Additionally, each of network elements 102a-102c, server 104, and cloud services 106 may include a processor (e.g., CPU 112) that can execute software or an algorithm to perform activities as discussed herein. A processor can execute any type of instructions associated with the data to achieve the operations detailed herein. In one example, the processors could transform an element or an article (e.g., data) from one state or thing to another state or thing. In another example, the activities outlined herein may be implemented with fixed logic or programmable logic (e.g., software/computer instructions executed by a processor) and the elements identified herein could be some type of a programmable processor, programmable digital logic (e.g., a field programmable gate array

(FPGA), an EPROM, an EEPROM) or an ASIC that includes digital logic, software, code, electronic instructions, or any suitable combination thereof. Any of the potential processing elements, modules, and machines described herein should be construed as being encompassed within the broad term 'processor.'

[0048] Turning to FIGURE 2, FIGURE 2 is a simplified block diagram of a portion of a system 100 to help facilitate exploration of an application in accordance with an embodiment of the present disclosure. In an example, network element 102b can include memory 110, CPU 112, security engine 114, and one or more applications 116a and 116b. Memory 110 can include one or more files 118a-118c and screen signature database 120. Screen signature database 120 can include one or more screen signatures 128a and 128b. Each screen signature can include an associated exploration strategy. For example, screen signature 128a includes or is associated with exploration strategy 130a and screen signature 128b includes or is associated with exploration strategy 130b. Each screen signature 128a and 128b can be a screen signature or identifier of a specific image. Each exploration strategy is an exploration strategy that can be used to explore an image with a specific screen signature. For example, if a signature of a target image matches or is similar to screen signature 128a, then exploration strategy 130a can be used to explore the target image. Each exploration strategy 130a and 130b can be a series or set of events such as a taping on a certain area of the screen that includes the image, scrolling the image to the left or the right, scrolling the image up or down, inputting a sequence of letters, numbers, and/or symbols (e.g., 1_dummy_email@dummy_server23.com), waiting for five (5) seconds, etc.). For example, exploration strategy 130a may be to enter data into a user interface element and then selecting a "next" user interface element (e.g., a "next" icon).

[0049] Security engine 114 can include a signature engine 132 and a similarity engine 134. Signature engine 132 can be configured to create a screen signature of a target image. The screen signature can be an abstracted mathematical representation of a target image. Similarity engine 134 can be configured to use the screen signature created by the signature engine 132 to search screen signature database 120 and to identify similar images with a similar screen signature. Once a similar screen signature is found in screen signature database 120, an exploration strategy associated with the similar screen signature can be executed or performed on the target image.

[0050] Turning to FIGURE 3, FIGURE 3 is a block diagram of a system to help facilitate a system and method for application exploration, in accordance with an embodiment. FIGURE 3 illustrates two different images that may be displayed on a display (e.g., display 122). In an example, a first screen image 136 may be part of a first application (e.g., application 116a) and has nine square user interface elements with the numbers 1-9 in consecutive order starting at the top most left box. A second screen image 138 may be part of a second application (e.g., application 116b) and has nine round user interface elements with the numbers 1-3 starting on the bottom row, then 4-6 on the middle row, and 7-9 on the top row. Current application exploration systems would determine that the images are different as the numbers are in a different order and the boxes, buttons, or user interface elements are different. However, the same exploration strategy could be used to explore each of first screen image 136 and second screen image 138.

[0051] Signature engine 132 (illustrated in FIGURE 2) can be configured to create a screen signature (e.g., an abstracted representation) of first screen image 136 and second screen image 138. For example, signature engine 132 can create a first screen signature 140 from first screen image 136 and a second screen signature 142 from second screen image 138. As illustrated in FIGURE 3, first screen signature 140 is similar to second screen signature 142 and the same exploration strategy could be used to explore each screen image. For example, an exploration strategy can be executed where the user interface elements are selected in one possible combination or order (e.g., the top row user interface elements are selected) or the user interface elements are selected in two or more different possible combinations orders.

[0052] Turning to FIGURES 4A and 4B, FIGURES 4A and 4B are a block diagram of a system to help facilitate a system and method for application exploration, in accordance with an embodiment. In an example, as illustrated in FIGURE 4A, an explored screen image 144 has user interface elements where a mobile number is to be entered and a next icon that can be selected. Signature engine 132 (illustrated in FIGURE 2) can be configured to create an explored screen signature 146, where explored screen signature 146 is an abstracted representation of explored screen image 144. Explored screen signature 146 can be stored in screen signature database 120 (illustrated in FIGURE 2) as a signature (e.g., screen signature 128a illustrated in FIGURE 2). An exploration strategy (e.g., exploration strategy 130a

illustrated in FIGURE 2) can be created for explored screen image 144 and associated with explored screen signature 146. For example, the exploration strategy may include entering data into the user interface elements and selecting the "next" user interface element (the "next" icon).

[0053] In an example, as illustrated in FIGURE 4B, a target screen image 148 may be part of an application that needs to be explored. Target screen image 148 can be captured and analyzed by signature engine 132 to create a target screen signature 150. Target screen signature 150 can be a signature of target screen image 148. Similarity engine 134 (illustrated in FIGURE 2) can compare target screen signature 150 to signatures stored in screen signature database 120 (illustrated in FIGURE 2) to try and determine one or more similar screen signatures. Because target screen signature 150 is similar to explored screen signature 146, a match can be identified and the exploration strategy associated with explored screen signature 146 can be used to explore target screen image 148. For example, the exploration strategy may be used to enter data into the user interface elements and select the "next" user interface element.

[0054] Turning to FIGURE 5, FIGURE 5 is a block diagram of a system to help facilitate a system and method for application exploration, in accordance with an embodiment. FIGURE 5 illustrates a third screen signature 154 and a fourth screen signature 156 that may be abstracted from two different images from two different webpages. In an example, third screen signature 154 can be an abstracted image from a webpage that allows viewing of video (e.g., YouTube) and fourth screen signature 156 can be an abstracted image from a webpage that allows viewing of news and/or headline related content (e.g., a newspaper's website). When third screen signature 154 is compared with fourth screen signature 156, a comparison result image 158 can be generated. As illustrated by comparison results image 158, the two images from the two different webpages are not similar and the same exploration strategy could not be used on both images. More specifically, an exploration strategy for a screen image that displays a video image would be very different than an exploration strategy for a screen image that includes news and/or headlines.

[0055] Turning to FIGURE 6, FIGURE 6 is an example flowchart illustrating possible operations of a flow 600 that may be associated with a system and method for application exploration, in accordance with an embodiment. In an embodiment, one or more operations

of flow 600 may be performed by security engine 114, signature engine 132, and/or similarity engine 134. At 602, a displayed image from an application is captured. For example, a screenshot of the image may be generated. The displayed image can be part of an application. At 604, a signature of the captured image is created. At 606, a database is queried for a matching or similar signature as the signature of the captured image. At 608, the system determines if there was a matching or similar signature as the signature of the displayed image. If there was not a matching or similar signature as the signature of the displayed image, then a message can be communicated to an administrator that no match was found, as in 610. In an example, the message that no match was found may be part of a generated report that is communicated to an administrator after a period of time, a number of displayed images have been analyzed, a number of applications have been analyzed, or some other event. If there was a matching or similar signature as the signature of the displayed image, then an exploration strategy associated with the matching or similar signature of the displayed image is acquired, as in 612. At 614, the exploration strategy is performed on the displayed image from the application. The exploration strategy allows the system to explore at least a portion of the application that caused the image to be displayed on the display.

[0056] Turning to FIGURE 7, FIGURE 7 is an example flowchart illustrating possible operations of a flow 700 that may be associated with a system and method for application exploration, in accordance with an embodiment. In an embodiment, one or more operations of flow 700 may be performed by security engine 114, signature engine 132, and/or similarity engine 134. At 702, a displayed image from an application is captured. For example, a screenshot of the image may be generated. The displayed image can be part of an application. At 704, a signature of the captured image is created. At 706, a database is queried for a matching or similar signature as the signature of the captured image. At 708, the system determines if there was a matching or similar signature as the signature of the captured image. If there was not a matching or similar signature as the signature of the captured image, then an exploration strategy for the displayed image from the application is created, as in 710. At 712, the exploration strategy is added to the database. At 714, the exploration strategy is performed on the displayed image from the application. If there was a matching or similar signature as the signature of the captured image, then an exploration strategy associated with the matching or similar signature of the captured image is acquired, as in 716.

At 714, the exploration strategy is performed on the displayed image from the application. The exploration strategy allows the system to explore at least a portion of the application that caused the image to be displayed on the display.

[0057] Turning to FIGURE 8, FIGURE 8 is an example flowchart illustrating possible operations of a flow 800 that may be associated with a system and method for application exploration, in accordance with an embodiment. In an embodiment, one or more operations of flow 800 may be performed by security engine 114 signature engine 132, and/or similarity engine 134. At 802, a screenshot of a displayed image from an application is acquired. In an example, the displayed image from the application is acquired by some means other than by a screenshot. At 804, a signature of the screenshot is created. At 806, a database is queried for a matching or similar signature as the signature of the screenshot. At 808, the system determines if there was a matching or similar signature as the signature of the screenshot. If there was a matching or similar signature as the signature of the screenshot, then an exploration strategy associated with the matching or similar signature of the screenshot is acquired, as in 810. At 812, the exploration strategy is performed on the displayed image from the application. At 814, the system determines if a new displayed image was generated during the exploration strategy. If a new displayed image was generated during the exploration strategy, then the system returns to 802, where a screenshot of the (new) displayed image from the application is acquired. If a new displayed image was not generated during the exploration strategy, then the process ends. If there was not a matching or similar signature as the signature of the screenshot, then an exploration strategy for the displayed image from the application is created, as in 816. At 818, the exploration strategy is added to the database. At 812, the exploration strategy is performed on the displayed image from the application and the system determines if a new displayed image was generated during the exploration strategy, as in 814. The exploration strategy of the target screen and any new images generated during execution of the exploration strategy allows the system to explore at least a portion of the application that caused the image to be displayed on the display.

[0058] Turning to FIGURE 9, FIGURE 9 is an example flowchart illustrating possible operations of a flow 900 that may be associated with a system and method for application exploration, in accordance with an embodiment. In an embodiment, one or more operations of flow 900 may be performed by security engine 114, signature engine 132, and/or similarity

engine 134. At 902, a displayed image from an application is identified. In an example, the displayed image from the application may be captured (e.g., the displayed image from the application may be captured by using a screenshot of the image or some other means). At 904, the displayed image from the application is analyzed to identify a user interface element. At 906, the system determines if a user interface element was detected. If a user interface element was detected, then the detected user interface element is added to an exploration strategy for the displayed image, as in 908 and the system returns to 904 where the displayed image from the application is analyzed to identify a new user interface element. At 906, the system determines if a (new) user interface element was detected. If a (new) user interface element was detected, then the detected (new) user interface element is added to an exploration strategy for the displayed image, as in 908 and the system returns to 904. The process ends when a user interface element (or new user interface elements) is not identified.

[0059] Turning to FIGURE 10, FIGURE 10 is an example flowchart illustrating possible operations of a flow 1000 that may be associated with a system and method for application exploration, in accordance with an embodiment. In an embodiment, one or more operations of flow 1000 may be performed by security engine 114, signature engine 132, and/or similarity engine 134. At 1002, a displayed image from an application is identified. In an example, the displayed image from the application may be captured (e.g., the displayed image from the application may be captured by using a screenshot of the image or some other means). At 1004, a user interface element in the displayed image from the application is identified. At 1006, data is entered into the user interface element. At 1008, the result of the data entered into the user interface element is recorded. For example, the result of the data being entered into the user interface element can be recorded in an application exploration report. The application exploration report can be used by an administrator to determine if an application includes some type of malicious activity, the application has privacy issues, the application may have navigation issues for a user, the application is benign, etc. At 1010, the system determines if a new image was generated. If a new image was not generated, then the system determines if there is another user interface element, as in 1012. If there is another user interface element, then the system returns to 1004 and a (new) user interface element in the displayed image from the application is identified. If a new image was generated, then the new image is separately analyzed, as in 1014 and the system determines

if there is another user interface element, as in 1012. For example, based on the data entered into the user interface element, the application may cause a new image to be generated. In a specific example, the new image can be analyzed using one of flows illustrated in FIGURE 6-8.

[0060] Turning to FIGURE 11, FIGURE 11 illustrates a computing system 1100 that is arranged in a point-to-point (PtP) configuration according to an embodiment. In particular, FIGURE 11 shows a system where processors, memory, and input/output devices are interconnected by a number of point-to-point interfaces. Generally, one or more of the network elements of system 100 may be configured in the same or similar manner as computing system 1100.

[0061] As illustrated in FIGURE 11, system 1100 may include several processors, of which only two, processors 1102a and 1102b, are shown for clarity. While two processors 1102a and 1102b are shown, it is to be understood that an embodiment of system 1100 may also include only one such processor. Processors 1102a and 1102b may each include a set of cores (i.e., processors cores 1104a and 1104b and processors cores 1104c and 1104d) to execute multiple threads of a program. The cores may be configured to execute instruction code in a manner similar to that discussed above with reference to FIGURES 1-10. Each processor 1102a and 1102b may include at least one shared cache 1106a and 1106b respectively. Shared caches 1106a and 1106b may each store data (e.g., instructions) that are utilized by one or more components of processors 1102a and 1102b, such as processor cores 1104a and 1104b of processor 1102a and processor cores 1104c and 1104d of processor 1102b.

[0062] Processors 1102a and 1102b may also each include integrated memory controller logic (MC) 1108a and 1108b respectively to communicate with memory elements 1110a and 1110b. Memory elements 1110a and/or 1110b may store various data used by processors 1102a and 1102b. In alternative embodiments, memory controller logic 1108a and 1108b may be discrete logic separate from processors 1102a and 1102b.

[0063] Processors 1102a and 1102b may be any type of processor and may exchange data via a point-to-point (PtP) interface 1112 using point-to-point interface circuits 1114a and 1114b respectively. Processors 1102a and 1102b may each exchange data with a chipset 1116 via individual point-to-point interfaces 1118a and 1118b using point-to-point interface

circuits 1120a-1120d. Chipset 1116 may also exchange data with a high-performance graphics circuit 1122 via a high-performance graphics interface 1124, using an interface circuit 1126, which could be a PtP interface circuit. In alternative embodiments, any or all of the PtP links illustrated in FIGURE 11 could be implemented as a multi-drop bus rather than a PtP link.

[0064] Chipset 1116 may be in communication with a bus 1128 via an interface circuit 1130. Bus 1128 may have one or more devices that communicate over it, such as a bus bridge 1132 and I/O devices 1134. Via a bus 1136, bus bridge 1132 may be in communication with other devices such as a keyboard/mouse 1138 (or other input devices such as a touch screen, trackball, etc.), communication devices 1140 (such as modems, network interface devices, or other types of communication devices that may communicate through a network), audio I/O devices 1142, and/or a data storage device 1144. Data storage device 1144 may store code 1146, which may be executed by processors 1102a and/or 1102b. In alternative embodiments, any portions of the bus architectures could be implemented with one or more PtP links.

[0065] The computer system depicted in FIGURE 11 is a schematic illustration of an embodiment of a computing system that may be utilized to implement various embodiments discussed herein. It will be appreciated that various components of the system depicted in FIGURE 11 may be combined in a system-on-a-chip (SoC) architecture or in any other suitable configuration. For example, embodiments disclosed herein can be incorporated into systems including mobile devices such as smart cellular telephones, tablet computers, personal digital assistants, portable gaming devices, etc. It will be appreciated that these mobile devices may be provided with SoC architectures in at least some embodiments.

[0066] Turning to FIGURE 12, FIGURE 12 is a simplified block diagram associated with an example ecosystem SOC 1200 of the present disclosure. At least one example implementation of the present disclosure can include the device pairing in a local network features discussed herein and an advanced reduced instruction set computer machine (ARM) component. For example, the example of FIGURE 12 can be associated with any ARM core (e.g., A-9, A-15, etc.). Further, the architecture can be part of any type of tablet, smartphone (inclusive of Android™ phones, iPhones™), iPad™, Google Nexus™, Microsoft Surface™, personal computer, server, video processing components, laptop computer (inclusive of any type of notebook), Ultrabook™ system, any type of touch-enabled input device, etc.

[0067] In this example of FIGURE 12, ecosystem SOC 1200 may include multiple cores 1202a and 1202b, an L2 cache control 1204, a graphics processing unit (GPU) 1206, a video codec 1208, a liquid crystal display (LCD) I/F 1210 and an interconnect 1212. L2 cache control 1204 can include a bus interface unit 1214, a L2 cache 1216. Liquid crystal display (LCD) I/F 1210 may be associated with mobile industry processor interface (MIPI)/ high-definition multimedia interface (HDMI) links that couple to an LCD.

[0068] Ecosystem SOC 1200 may also include a subscriber identity module (SIM) I/F 1218, a boot read-only memory (ROM) 1220, a synchronous dynamic random-access memory (SDRAM) controller 1222, a flash controller 1224, a serial peripheral interface (SPI) master 1228, a suitable power control 1230, a dynamic RAM (DRAM) 1232, and flash 1234. In addition, one or more embodiments include one or more communication capabilities, interfaces, and features such as instances of Bluetooth™ 1236, a 3G modem 1238, a global positioning system (GPS) 1240, and an 802.11 Wi-Fi 1242.

[0069] In operation, the example of FIGURE 12 can offer processing capabilities, along with relatively low power consumption to enable computing of various types (e.g., mobile computing, high-end digital home, servers, wireless infrastructure, etc.). In addition, such an architecture can enable any number of software applications (e.g., Android™, Adobe® Flash® Player, Java Platform Standard Edition (Java SE), JavaFX, Linux, Microsoft Windows Embedded, Symbian and Ubuntu, etc.). In at least one example embodiment, the core processor may implement an out-of-order superscalar pipeline with a coupled low-latency level-2 cache.

[0070] Turning to FIGURE 13, FIGURE 13 illustrates a processor core 1300 according to an embodiment. Processor core 1300 may be the core for any type of processor, such as a micro-processor, an embedded processor, a digital signal processor (DSP), a network processor, or other device to execute code. Although only one processor core 1300 is illustrated in Figure 13, a processor may alternatively include more than one of the processor core 1300 illustrated in Figure 13. For example, processor core 1300 represents one example embodiment of processors cores 1104a-1104d shown and described with reference to processors 1102a and 1102b of FIGURE 11. Processor core 1300 may be a single-threaded core or, for at least one embodiment, processor core 1300 may be multithreaded in that it may include more than one hardware thread context (or "logical processor") per core.

[0071] FIGURE 13 also illustrates a memory 1302 coupled to processor core 1300 in accordance with an embodiment. Memory 1302 may be any of a wide variety of memories (including various layers of memory hierarchy) as are known or otherwise available to those of skill in the art. Memory 1302 may include code 1304, which may be one or more instructions, to be executed by processor core 1300. Processor core 1300 can follow a program sequence of instructions indicated by code 1304. Each instruction enters a front-end logic 1306 and is processed by one or more decoders 1308. The decoder may generate, as its output, a micro operation such as a fixed width micro operation in a predefined format, or may generate other instructions, microinstructions, or control signals that reflect the original code instruction. Front-end logic 1306 also includes register renaming logic 1310 and scheduling logic 1312, which generally allocate resources and queue the operation corresponding to the instruction for execution.

[0072] Processor core 1300 can also include execution logic 1314 having a set of execution units 1316-1 through 1316-N. Some embodiments may include a number of execution units dedicated to specific functions or sets of functions. Other embodiments may include only one execution unit or one execution unit that can perform a particular function. Execution logic 1314 performs the operations specified by code instructions.

[0073] After completion of execution of the operations specified by the code instructions, back-end logic 1318 can retire the instructions of code 1304. In one embodiment, processor core 1300 allows out of order execution but requires in order retirement of instructions. Retirement logic 1320 may take a variety of known forms (e.g., re-order buffers or the like). In this manner, processor core 1300 is transformed during execution of code 1304, at least in terms of the output generated by the decoder, hardware registers and tables utilized by register renaming logic 1310, and any registers (not shown) modified by execution logic 1314.

[0074] Although not illustrated in FIGURE 13, a processor may include other elements on a chip with processor core 1300, at least some of which were shown and described herein with reference to FIGURE 11. For example, as shown in FIGURE 11, a processor may include memory control logic along with processor core 1300. The processor may include I/O control logic and/or may include I/O control logic integrated with memory control logic.

[0075] Note that with the examples provided herein, interaction may be described in terms of two, three, or more network elements. However, this has been done for purposes of clarity and example only. In certain cases, it may be easier to describe one or more of the functionalities of a given set of flows by only referencing a limited number of network elements. It should be appreciated that system 100 and its teachings are readily scalable and can accommodate a large number of components, as well as more complicated/sophisticated arrangements and configurations. Accordingly, the examples provided should not limit the scope or inhibit the broad teachings of system 100 as potentially applied to a myriad of other architectures.

[0076] It is also useful to note that the operations in the preceding flow diagrams (i.e., FIGURES 6-10-) illustrate only some of the possible correlating scenarios and patterns that may be executed by, or within, system 100. Some of these operations may be deleted or removed where appropriate, or these operations may be modified or changed considerably without departing from the scope of the present disclosure. In addition, a number of these operations have been described as being executed concurrently with, or in parallel to, one or more additional operations. However, the timing of these operations may be altered considerably. The preceding operational flows have been offered for purposes of example and discussion. Substantial flexibility is provided by system 100 in that any suitable arrangements, chronologies, configurations, and timing mechanisms may be provided without departing from the teachings of the present disclosure.

[0077] Although the present disclosure has been described in detail with reference to particular arrangements and configurations, these example configurations and arrangements may be changed without departing from the scope of the present disclosure. Moreover, certain components may be combined, separated, reduced, eliminated, or added based on particular needs and implementations. Additionally, although system 100 has been illustrated with reference to particular elements and operations that facilitate the communication process, these elements and operations may be replaced by any suitable architecture, protocols, and/or processes that achieve the intended functionality of system 100

[0078] Numerous other changes, substitutions, variations, alterations, and modifications may be ascertained to one skilled in the art and it is intended that the present

disclosure encompass all such changes, substitutions, variations, alterations, and modifications as falling within the scope of the appended claims. In order to assist the United States Patent and Trademark Office (USPTO) and, additionally, any readers of any patent issued on this application in interpreting the claims appended hereto, Applicant wishes to note that the Applicant: (a) does not intend any of the appended claims to invoke paragraph six (6) of 35 U.S.C. section 112 as it exists on the date of the filing hereof unless the words "means for" or "step for" are specifically used in the particular claims; and (b) does not intend, by any statement in the specification, to limit this disclosure in any way that is not otherwise reflected in the appended claims.

OTHER NOTES AND EXAMPLES

[0079] Example M1 is a method including capturing an image on a display, where the image includes one or more user interface elements and is part of an application, creating a screen signature of the image, determining an exploration strategy for the image based on the screen signature, and performing the exploration strategy on the image.

[0080] In Example M2, the subject matter of Example M1 can optionally include where the exploration strategy includes interacting with each of the one or more user interface elements.

[0081] In Example M3, the subject matter of any one of the Examples M1-M2 can optionally include where the exploration strategy is operating system agnostic.

[0082] In Example M4, the subject matter of any one of the Examples M1-M3 can optionally where the image is abstracted to create the screen signature.

In Example M5, the subject matter of any one of the Examples M1-M4 can optionally include comparing the screen signature of the image to a plurality of screen signatures in a screen signature database and determining that a first of the plurality of screen signatures matches the screen signature for the image, wherein determining the exploration strategy for the image includes using a stored exploration strategy that is associated with the first of the plurality of screen signatures.

[0083] In Example M6, the subject matter of any one of the Examples M1-M5 can optionally include where the screen signature of the image matches two or more of the plurality of screen signatures, wherein each of the two or more of the plurality of screen

signatures is associated with a different exploration strategy and each of the different exploration strategies is performed on the image.

[0084] In Example M7, the subject matter of any one of the Examples M1-M6 can optionally include where a new image is created when the exploration strategy is performed on the image.

[0085] In Example M8, the subject matter of any one of the Examples M1-M7 can optionally include creating a second screen signature of the new image, determining a second exploration strategy for the new image based on the second screen signature, and performing the second exploration strategy on the new image

[0086] Example C1 is at least one machine readable storage medium having one or more instructions that when executed by at least one processor, cause the at least one processor to capture an image on a display, where the image includes at least one user interface element and is part of an application, create a screen signature of the image, determine an exploration strategy for the image based on the screen signature, and perform the exploration strategy on the image.

[0087] In Example C2, the subject matter of Example C1 can optionally include where the image is abstracted to create the screen signature.

[0088] In Example C3, the subject matter of any one of Examples C1-C2 can optionally include where the exploration strategy is operating system agnostic.

[0089] In Example C4, the subject matter of any one of Examples C1-C3 can optionally include where the one or more instructions that when executed by the at least one processor, further cause the processor to compare the screen signature of the image to a plurality of screen signatures in a screen signature database, determine that a specific screen signature from the plurality of screen signatures in the screen signature database matches the screen signature of the image, and determine the exploration strategy for the image by using a stored exploration strategy that is associated with the specific screen signature that matches the screen signature of the image.

[0090] In Example C5, the subject matter of any one of Examples C1-C4 can optionally include where a new image is created when the exploration strategy is performed on the image.

[0091] In Example C6, the subject matter of any one of Example C1-C5 can optionally include where the one or more instructions that when executed by the at least one processor, further cause the processor to create a second screen signature of the new image, determine a second exploration strategy for the new image based on the second screen signature, and perform the second exploration strategy on the new image.

[0092] In Example C7, the subject matter of any one of Examples C1-C6 can optionally include where the display is a virtual display.

[0093] In Example A1, an apparatus can include a security engine. The security engine can be configured to capture an image on a display, where the image includes at least one user interface element and is part of an application, create a screen signature of the image, determine an exploration strategy for the image based on the screen signature, and perform the exploration strategy on the image.

[0094] In Example, A2, the subject matter of Example A1 can optionally include where the image is abstracted to create the screen signature.

[0095] In Example A3, the subject matter of any one of Examples A1-A2 can optionally include where the exploration strategy is operating system agnostic.

In Example A4, the subject matter of any one of Examples A1-A3 can optionally include where the security engine is further configured to compare the screen signature of the image to a plurality of screen signatures in a screen signature database and determine that a first of the plurality of screen signatures matches the screen signature for the image, wherein the determined exploration strategy for the image is a stored exploration strategy that is associated with the first of the plurality of screen signatures.

[0096] In Example A5, the subject matter of any one of Examples A1-A4 can optionally include where a new image is created when the exploration strategy is performed on the image.

[0097] In Example A6, the subject matter of any one of Examples A1-A5 can optionally include where the security engine if further configured to create a second screen signature of the new image, determine a second exploration strategy for the new image based on the second screen signature, and perform the second exploration strategy on the new image.

[0098] In Example A7, the subject matter of any one of Examples A1-A6 can optionally include where the display is a virtual display.

[0099] Example S1 is a system to explore an application, the system including a computer processing unit, memory, and a security engine. The security engine can include a signature engine and a similarity engine. The signature engine can be configured to capture an image on a display, where the image includes one or more user interface elements and is part of an application and create a screen signature of the image. The similarity engine can be configured to compare the screen signature of the image to a plurality of screen signatures in a screen signature database, and determine that a first of the plurality of screen signatures matches the screen signature for the image, wherein an exploration strategy for the image is a stored exploration strategy that is associated with the first of the plurality of screen signatures.

[00100] In Example S2, the subject matter of Example S1 can optionally include where a new image is created when the exploration strategy is performed on the image, where the signature engine is further configured to create a second screen signature of the new image, and the similarity engine is further configured to determine a second exploration strategy for the new image based on the second screen signature, where the second exploration strategy is performed on the new image.

[00101] In Example S3, the subject matter of any of the Examples S1-S2 can optionally include where the image is abstracted to create the screen signature.

[00102] In Example S4, the subject matter of any of the Examples S1-S3 can optionally include where the exploration strategy includes interacting with each of the one or more user interface elements.

[00103] In Example S5, the subject matter of any of the Examples S1-S4 can optionally include where the exploration strategy is operating system agnostic.

[00104] Example AA1 is an electronic device including means for capturing an image on a display, where the image includes one or more user interface elements and is part of an application, means for creating a screen signature of the image, means for determining an exploration strategy for the image based on the screen signature, and means for performing the exploration strategy on the image.

[00105] In Example AA2, the subject matter of Example AA1 can optionally include where the exploration strategy includes interacting with each of the one or more user interface elements.

[00106] In Example AA3, the subject matter of any one of Examples AA1 - AA2 can optionally include where the exploration strategy is operating system agnostic.

[00107] In Example AA4, the subject matter of any one of Examples AA1-AA3 can optionally include where the image is abstracted to create the screen signature.

[00108] In Example AA5, the subject matter of any one of Examples AA1-AA4 can optionally include means for comparing the screen signature of the image to a plurality of screen signatures in a screen signature database, means for determining that a specific screen signature from the plurality of screen signatures in the screen signature database matches the screen signature of the image, and means for determining the exploration strategy for the image by using a stored exploration strategy that is associated with the specific screen signature that matches the screen signature of the image.

[00109] In Example AA6, the subject matter of any one of Examples AA1-AA5 can optionally include where the screen signature of the image matches two or more of screen signatures from the screen signature database, where each of the two or more of screen signatures from the screen signature database are associated with a different exploration strategy and each of the different exploration strategies are performed on the image.

[00110] In Example AA7, the subject matter of any one of Example AA1-AA6 can optionally include where a new image is created when the exploration strategy is performed on the image.

[00111] In Example AA8, the subject matter of any one of Example AA1-AA7 can optionally include means for creating a second screen signature of the new image, means for determining a second exploration strategy for the new image based on the second screen signature, and means for performing the second exploration strategy on the new image.

[00112] Example X1 is a machine-readable storage medium including machine-readable instructions to implement a method or realize an apparatus as in any one of the Examples A1-A7, AA1-AA8 or M1-M8. Example Y1 is an apparatus comprising means for performing of any of the Example methods M1-M8. In Example Y2, the subject matter of Example Y1 can optionally include the means for performing the method comprising a processor and a memory. In Example Y3, the subject matter of Example Y2 can optionally include the memory comprising machine-readable instructions.

CLAIMS:

1. A method comprising:

capturing an image on a display, wherein the image includes one or more user interface elements and is part of an application;

creating a screen signature of the image;

determining an exploration strategy for the image based on the screen signature; and

performing the exploration strategy on the image.

2. The method of Claim 1, wherein the exploration strategy includes interacting with each of the one or more user interface elements.

3. The method of Claim 1, wherein the exploration strategy is operating system agnostic.

4. The method of Claim 1, wherein the image is abstracted to create the screen signature.

5. The method of Claim 1, further comprising:

comparing the screen signature of the image to a plurality of screen signatures in a screen signature database; and

determining that a first of the plurality of screen signatures matches the screen signature for the image, wherein determining the exploration strategy for the image includes using a stored exploration strategy that is associated with the first of the plurality of screen signatures.

6. The method of Claim 5, wherein the screen signature of the image matches two or more of the plurality of screen signatures, wherein each of the two or more of the

plurality of screen signatures is associated with a different exploration strategy and each of the different exploration strategies is performed on the image.

7. The method of Claim 1, wherein a new image is created when the exploration strategy is performed on the image.

8. The method of Claim 7, further comprising:

creating a second screen signature of the new image;

determining a second exploration strategy for the new image based on the second screen signature; and

performing the second exploration strategy on the new image.

9. At least one computer-readable medium comprising one or more instructions that when executed by at least one processor, cause the at least one processor to:

capture an image on a display, wherein the image includes at least one user interface element and is part of an application;

create a screen signature of the image;

determine an exploration strategy for the image based on the screen signature; and

perform the exploration strategy on the image.

10. The at least one computer-readable medium of Claim 9, wherein the image is abstracted to create the screen signature.

11. The at least one computer-readable medium of Claim 9, wherein the exploration strategy is operating system agnostic.

12. The at least one computer-readable medium of Claim 9, further comprising one or more instructions that when executed by the at least one processor, further cause the processor to:

compare the screen signature of the image to a plurality of screen signatures in a screen signature database; and

determine that a first of the plurality of screen signatures matches the screen signature for the image, wherein the determined exploration strategy for the image is a stored exploration strategy that is associated with the first of the plurality of screen signatures.

13. The at least one computer-readable medium of Claim 9, wherein a new image is created when the exploration strategy is performed on the image.

14. The at least one computer-readable medium of Claim 13, further comprising one or more instructions that when executed by the at least one processor, further cause the processor to:

create a second screen signature of the new image;

determine a second exploration strategy for the new image based on the second screen signature; and

perform the second exploration strategy on the new image.

15. The at least one computer-readable medium of Claim 9, wherein the display is a virtual display.

16. A system to explore an application, the system comprising:

memory, wherein the memory includes a screen signature database; and

a security engine, wherein the security engine includes a signature engine and a similarity engine,

wherein the signature engine is configured to:

capture an image on a display, wherein the image includes one or more user interface elements and is part of an application; and

create a screen signature of the image;

wherein the similarity engine is configured to:

compare the screen signature of the image to a plurality of screen signatures in a screen signature database; and

determine that a first of the plurality of screen signatures matches the screen signature for the image, wherein an exploration strategy for the image is a stored exploration strategy that is associated with the first of the plurality of screen signatures.

17. The system of Claim 16, wherein a new image is created when the exploration strategy is performed on the image, wherein the signature engine is further configured to create a second screen signature of the new image, and the similarity engine is further configured to determine a second exploration strategy for the new image based on the second screen signature, wherein the second exploration strategy is performed on the new image.

18. The system of Claim 16, wherein the image is abstracted to create the screen signature.

19. The system of Claim 16, wherein the exploration strategy includes interacting with each of the one or more user interface elements.

20. The system of Claim 16, wherein the exploration strategy is operating system agnostic.

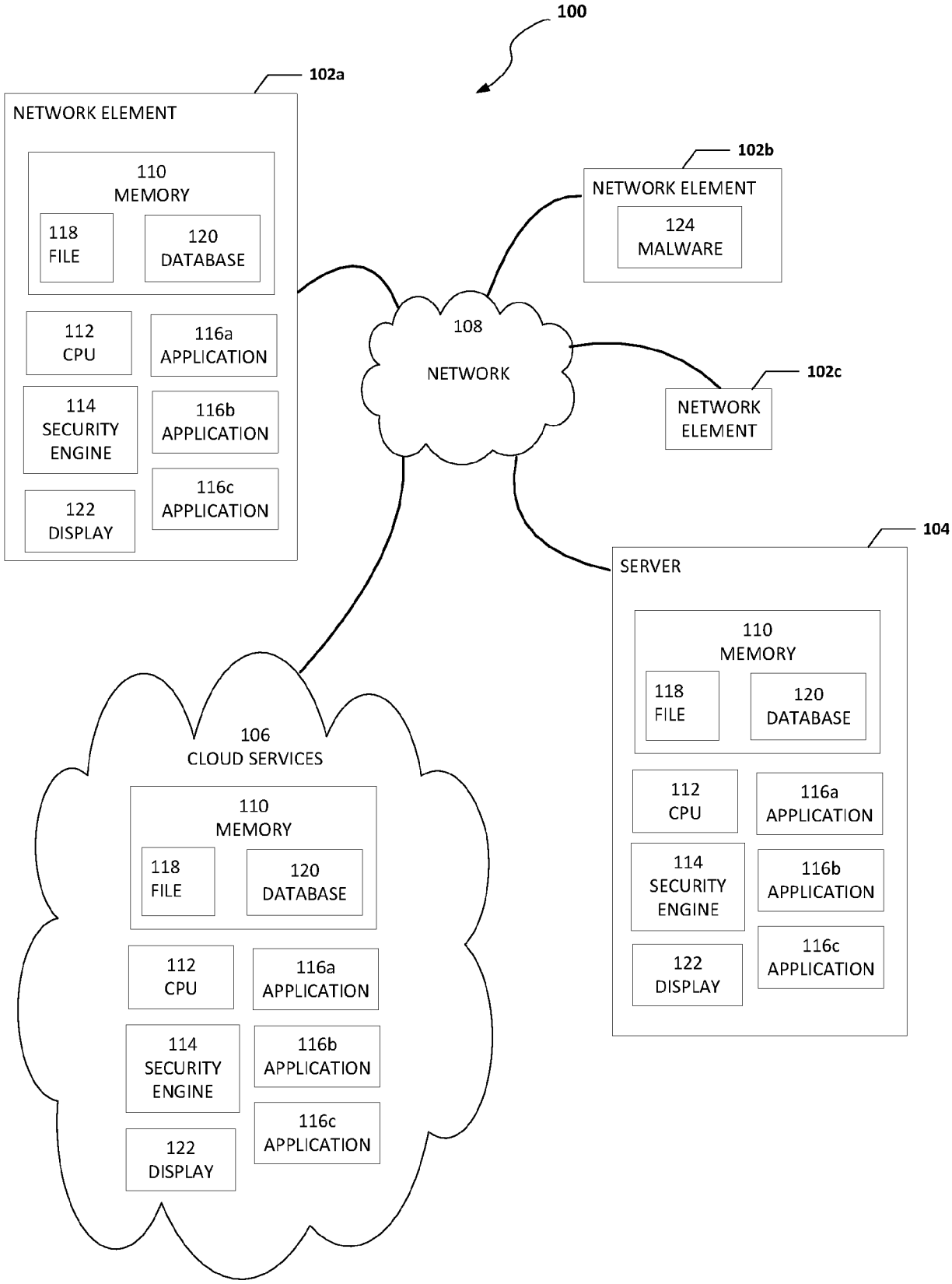

100
102a
NETWORK ELEMENT
110
MEMORY
118
FILE
120
DATABASE
112
CPU
116a
APPLICATION
114
SECURITY
ENGINE
116b
APPLICATION
122
DISPLAY
116c
APPLICATION
102b
NETWORK ELEMENT
124
MALWARE
108
NETWORK
102c
NETWORK
ELEMENT
104
SERVER
110
MEMORY
118
FILE
120
DATABASE
112
CPU
116a
APPLICATION
114
SECURITY
ENGINE
116b
APPLICATION
122
DISPLAY
116c
APPLICATION
106
CLOUD SERVICES
110
MEMORY
118
FILE
120
DATABASE
112
CPU
116a
APPLICATION
114
SECURITY
ENGINE
116b
APPLICATION
122
DISPLAY
116c
APPLICATION

FIGURE 1

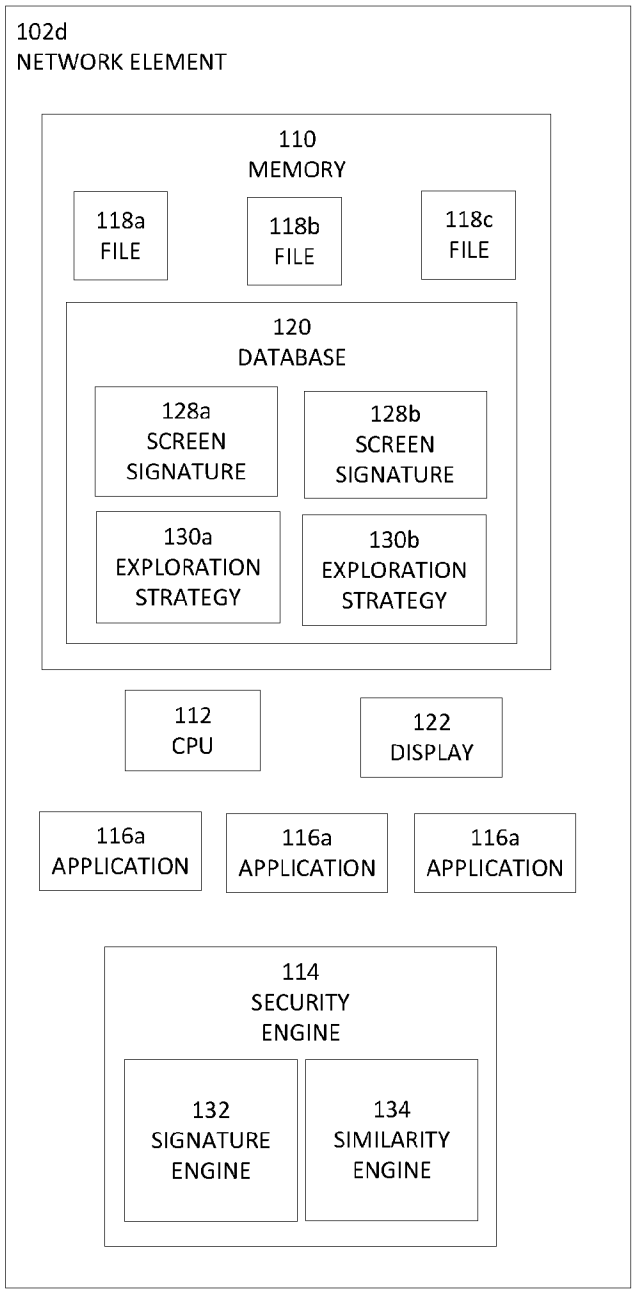

102d
NETWORK ELEMENT
110
MEMORY
118a
FILE
118b
FILE
118c
FILE
120
DATABASE
128a
SCREEN
SIGNATURE
128b
SCREEN
SIGNATURE
130a
EXPLORATION
STRATEGY
130b
EXPLORATION
STRATEGY
112
CPU
122
DISPLAY
116a
APPLICATION
116a
APPLICATION
116a
APPLICATION
114
SECURITY
ENGINE
132
SIGNATURE
ENGINE
134
SIMILARITY
ENGINE

FIGURE 2

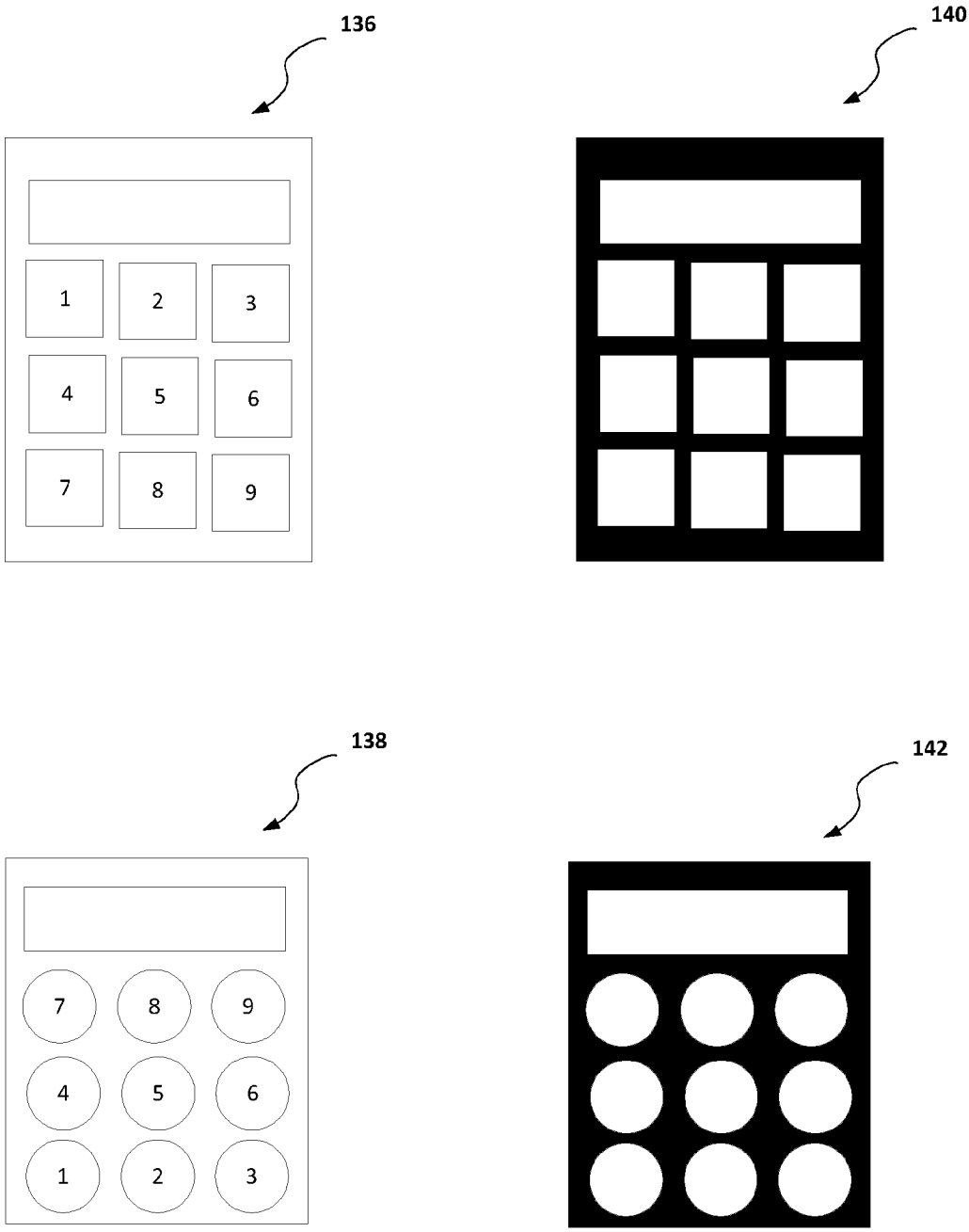


FIGURE 3

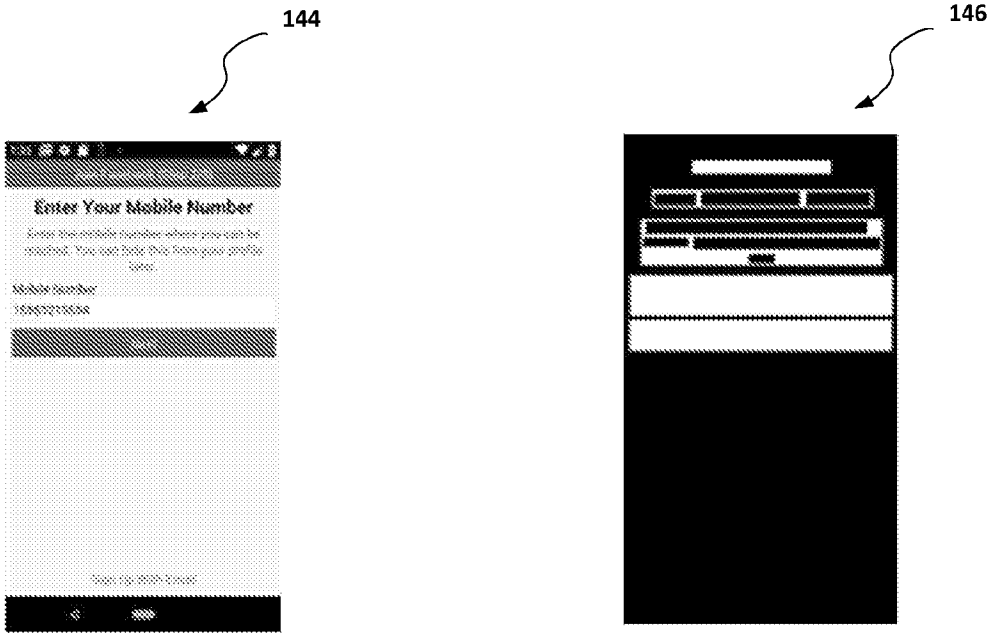


FIGURE 4A

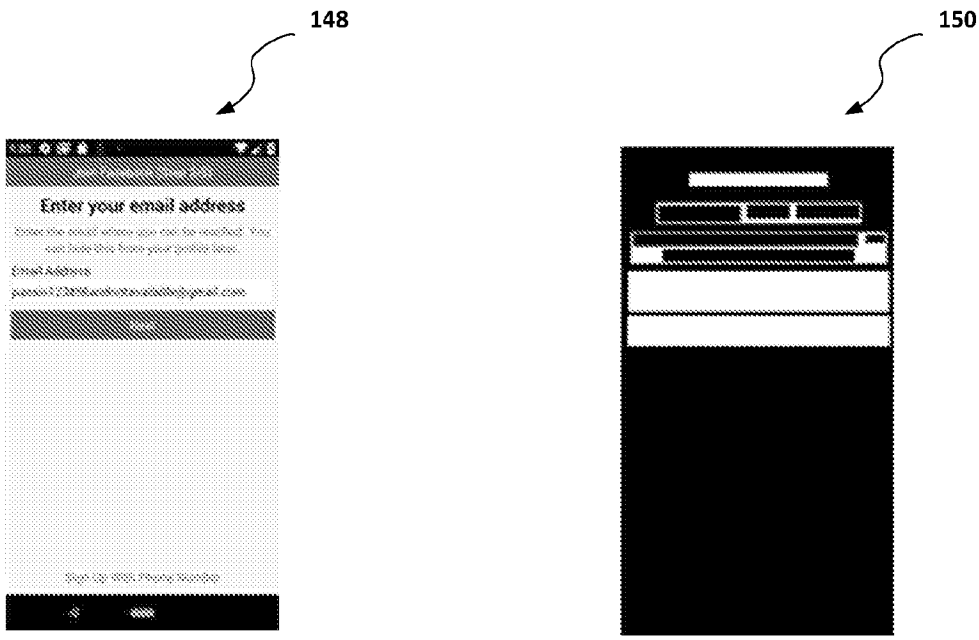


FIGURE 4B

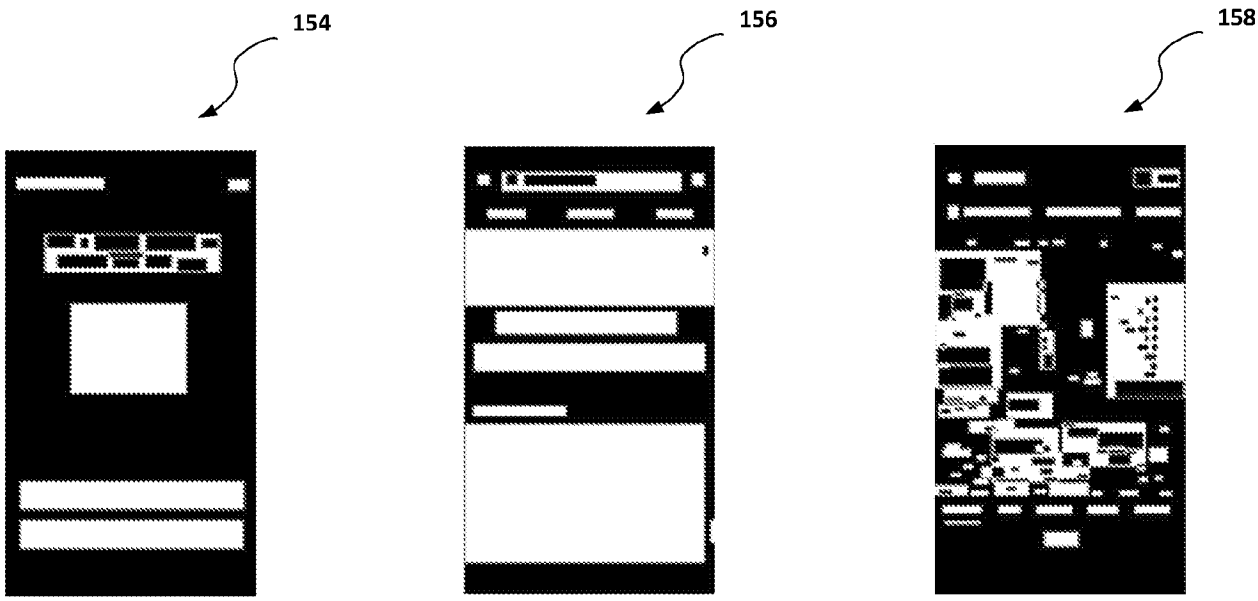


FIGURE 5

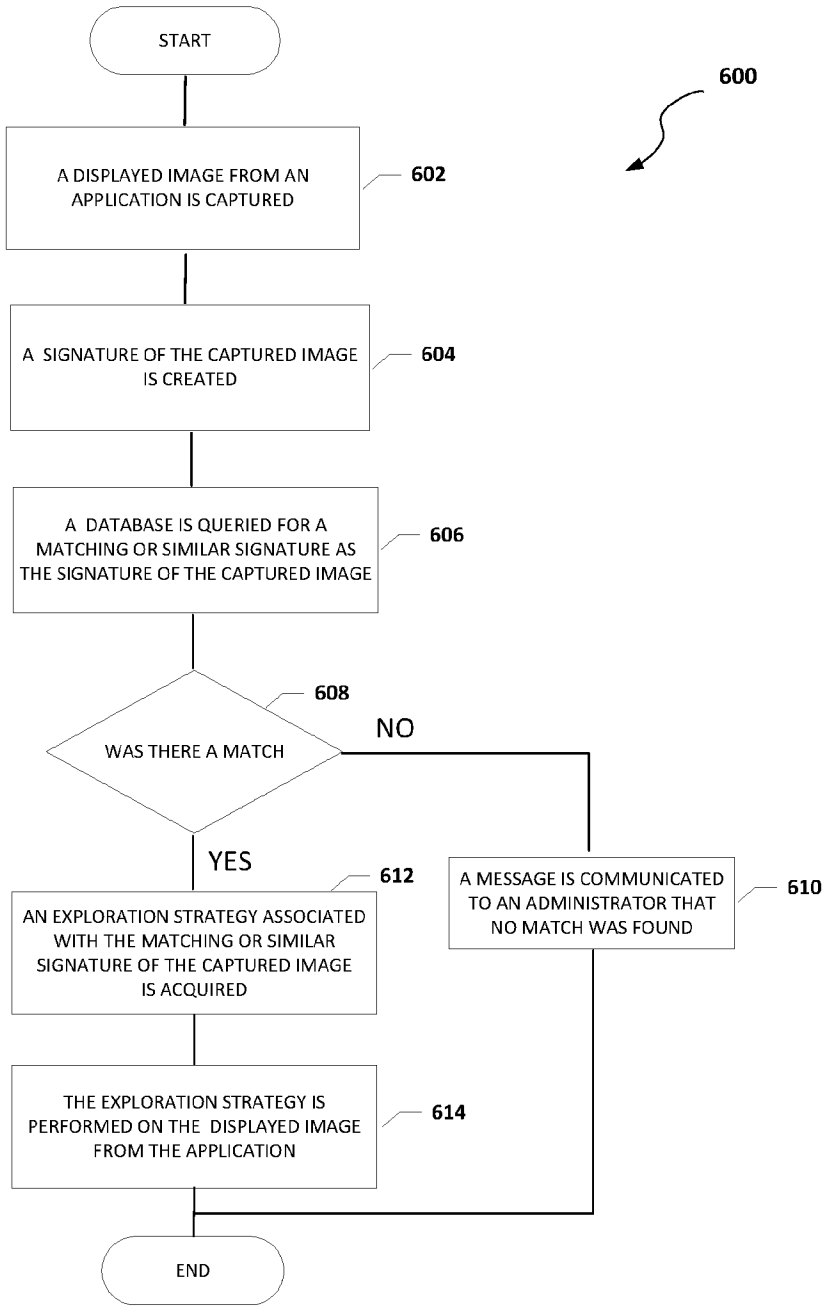

START
600
A DISPLAYED IMAGE FROM AN APPLICATION IS CAPTURED
602
A SIGNATURE OF THE CAPTURED IMAGE IS CREATED
604
A DATABASE IS QUERIED FOR A MATCHING OR SIMILAR SIGNATURE AS THE SIGNATURE OF THE CAPTURED IMAGE
606
608
WAS THERE A MATCH
NO
YES
612
AN EXPLORATION STRATEGY ASSOCIATED WITH THE MATCHING OR SIMILAR SIGNATURE OF THE CAPTURED IMAGE IS ACQUIRED
A MESSAGE IS COMMUNICATED TO AN ADMINISTRATOR THAT NO MATCH WAS FOUND
610
THE EXPLORATION STRATEGY IS PERFORMED ON THE DISPLAYED IMAGE FROM THE APPLICATION
614
END

FIGURE 6

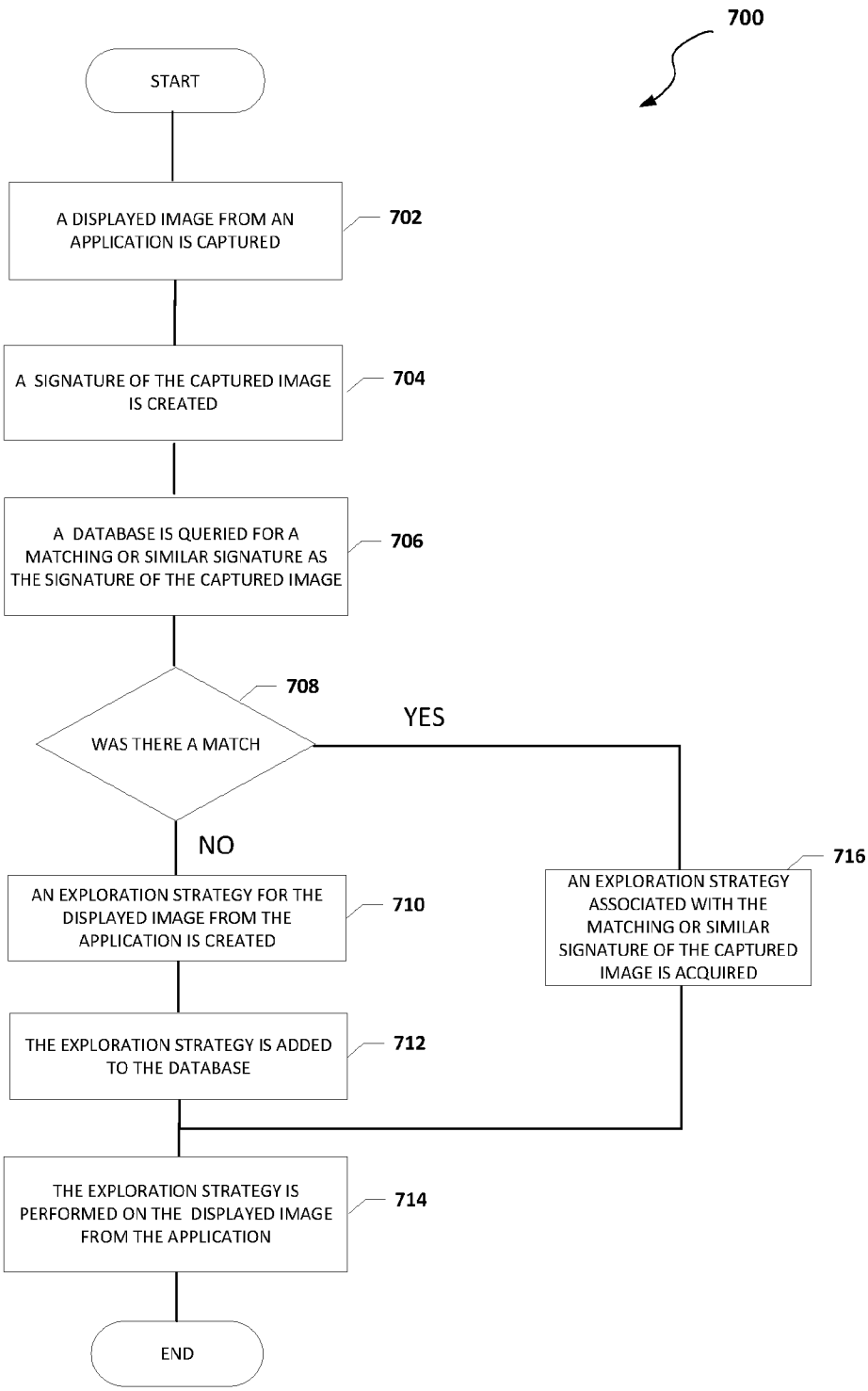

700
START
A DISPLAYED IMAGE FROM AN APPLICATION IS CAPTURED
702
A SIGNATURE OF THE CAPTURED IMAGE IS CREATED
704
A DATABASE IS QUERIED FOR A MATCHING OR SIMILAR SIGNATURE AS THE SIGNATURE OF THE CAPTURED IMAGE
706
708
WAS THERE A MATCH
YES
NO
AN EXPLORATION STRATEGY FOR THE DISPLAYED IMAGE FROM THE APPLICATION IS CREATED
710
AN EXPLORATION STRATEGY ASSOCIATED WITH THE MATCHING OR SIMILAR SIGNATURE OF THE CAPTURED IMAGE IS ACQUIRED
716
THE EXPLORATION STRATEGY IS ADDED TO THE DATABASE
712
THE EXPLORATION STRATEGY IS PERFORMED ON THE DISPLAYED IMAGE FROM THE APPLICATION
714
END

FIGURE 7

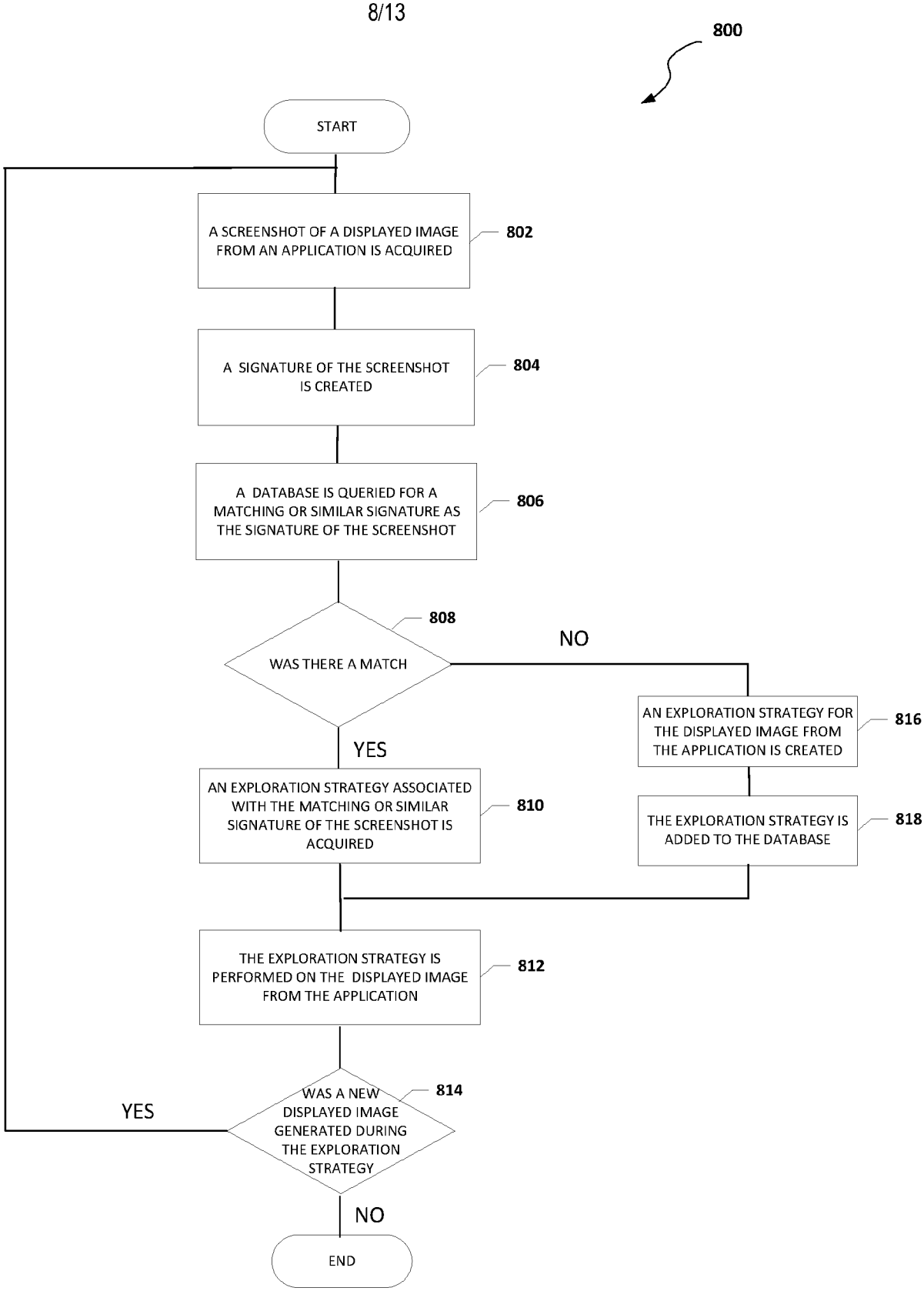

800
START
A SCREENSHOT OF A DISPLAYED IMAGE FROM AN APPLICATION IS ACQUIRED
802
A SIGNATURE OF THE SCREENSHOT IS CREATED
804
A DATABASE IS QUERIED FOR A MATCHING OR SIMILAR SIGNATURE AS THE SIGNATURE OF THE SCREENSHOT
806
808
WAS THERE A MATCH
NO
YES
AN EXPLORATION STRATEGY ASSOCIATED WITH THE MATCHING OR SIMILAR SIGNATURE OF THE SCREENSHOT IS ACQUIRED
810
AN EXPLORATION STRATEGY FOR THE DISPLAYED IMAGE FROM THE APPLICATION IS CREATED
816
THE EXPLORATION STRATEGY IS ADDED TO THE DATABASE
818
THE EXPLORATION STRATEGY IS PERFORMED ON THE DISPLAYED IMAGE FROM THE APPLICATION
812
WAS A NEW DISPLAYED IMAGE GENERATED DURING THE EXPLORATION STRATEGY
814
YES
NO
END

FIGURE 8

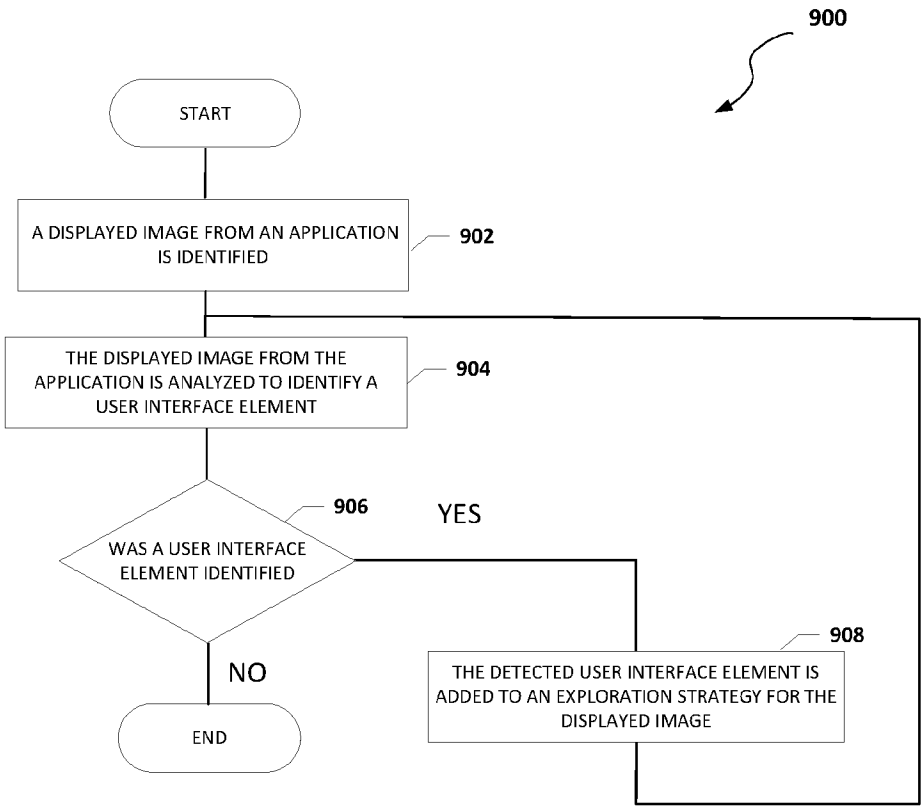

900
START
A DISPLAYED IMAGE FROM AN APPLICATION IS IDENTIFIED
902
THE DISPLAYED IMAGE FROM THE APPLICATION IS ANALYZED TO IDENTIFY A USER INTERFACE ELEMENT
904
WAS A USER INTERFACE ELEMENT IDENTIFIED
906
YES
NO
END
THE DETECTED USER INTERFACE ELEMENT IS ADDED TO AN EXPLORATION STRATEGY FOR THE DISPLAYED IMAGE
908

FIGURE 9

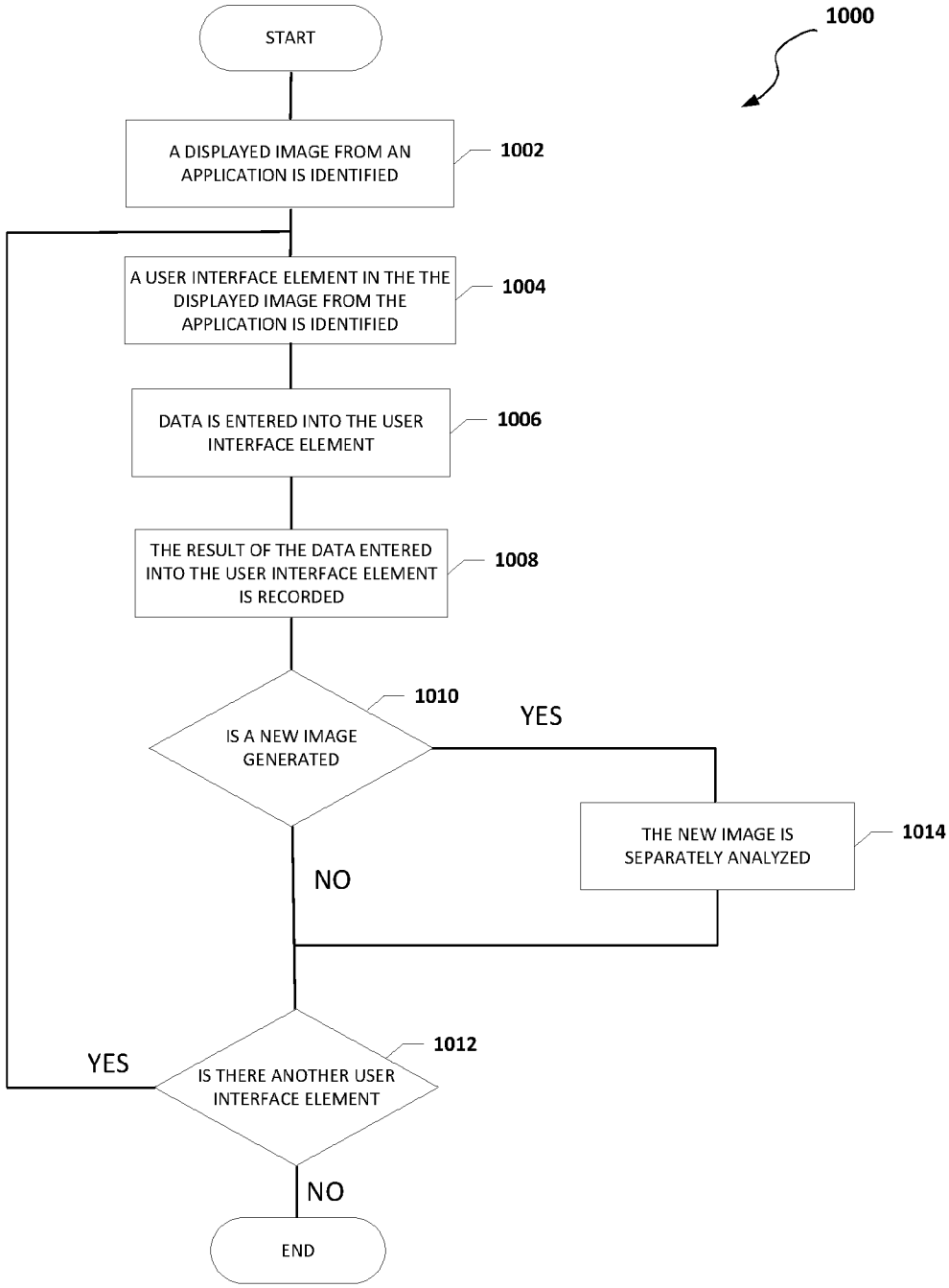

1000
START
A DISPLAYED IMAGE FROM AN APPLICATION IS IDENTIFIED
1002
A USER INTERFACE ELEMENT IN THE THE DISPLAYED IMAGE FROM THE APPLICATION IS IDENTIFIED
1004
DATA IS ENTERED INTO THE USER INTERFACE ELEMENT
1006
THE RESULT OF THE DATA ENTERED INTO THE USER INTERFACE ELEMENT IS RECORDED
1008
IS A NEW IMAGE GENERATED
1010
YES
THE NEW IMAGE IS SEPARATELY ANALYZED
1014
NO
IS THERE ANOTHER USER INTERFACE ELEMENT
1012
YES
NO
END

FIGURE 10

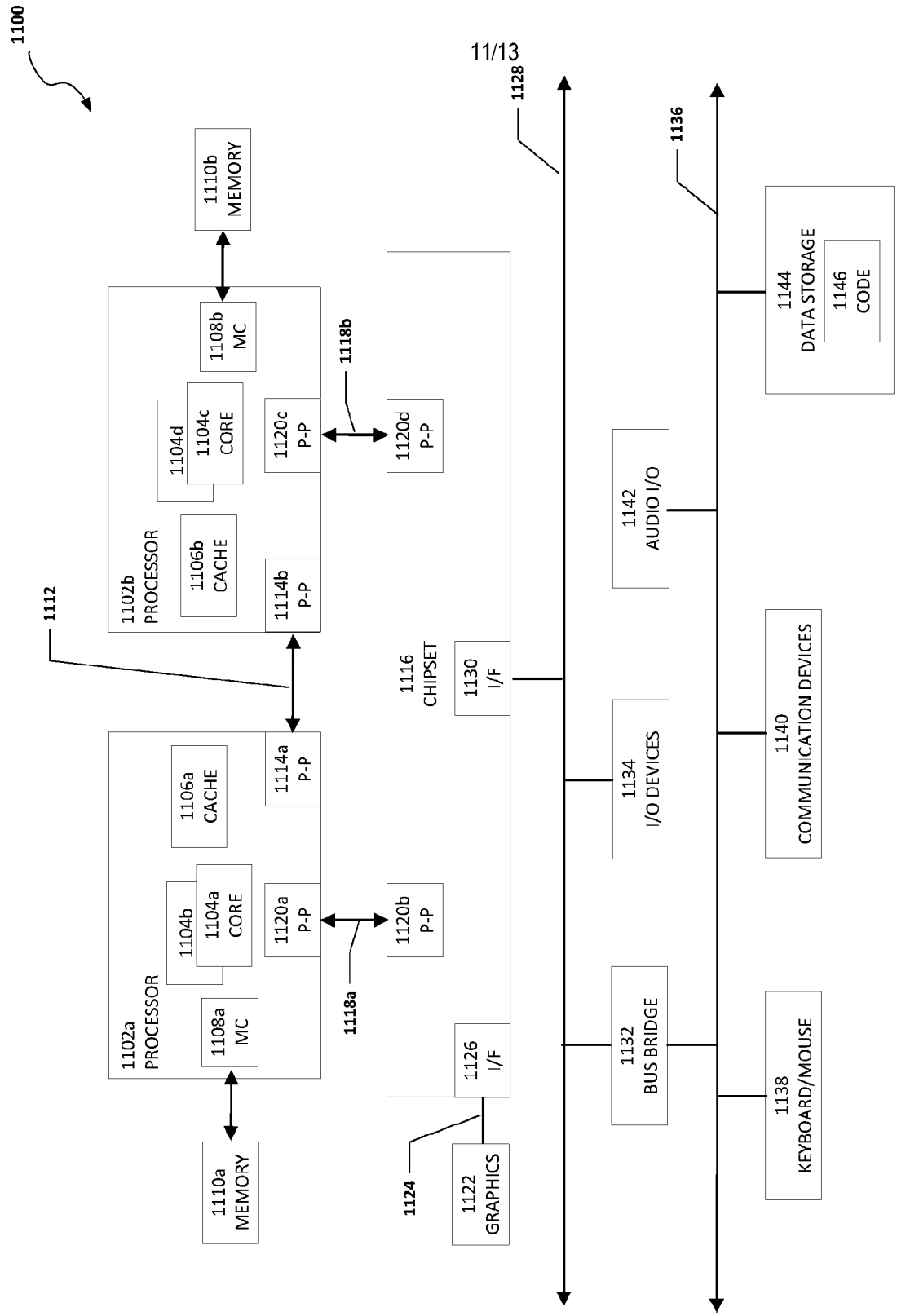

1100
11/13
1128
1136
1110b MEMORY
1108b MC
1104d
1104c CORE
1106b CACHE
1102b PROCESSOR
1120c P-P
1118b
1120d P-P
1114b P-P
1112
1116 CHIPSET
1130 I/F
1142 AUDIO I/O
1144 DATA STORAGE
1146 CODE
1106a CACHE
1114a P-P
1104b
1104a CORE
1102a PROCESSOR
1120a P-P
1118a
1120b P-P
1108a MC
1134 I/O DEVICES
1140 COMMUNICATION DEVICES
1126 I/F
1132 BUS BRIDGE
1138 KEYBOARD/MOUSE
1110a MEMORY
1124
1122 GRAPHICS

FIGURE 11

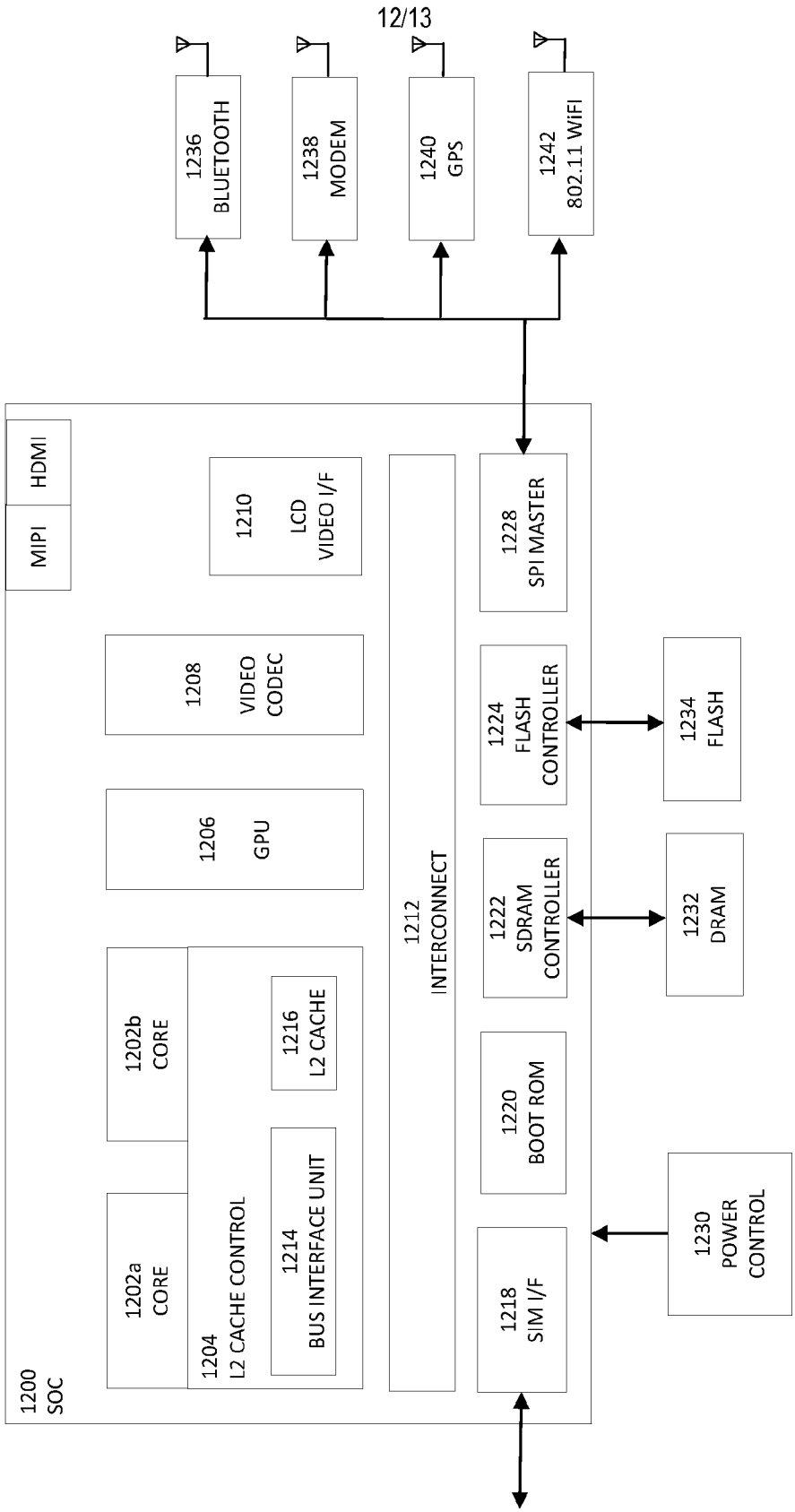

1236
BLUETOOTH
1238
MODEM
1240
GPS
1242
802.11 WiFi
1200
SOC
MIPI
HDMI
1210
LCD
VIDEO I/F
1208
VIDEO
CODEC
1206
GPU
1202a
CORE
1202b
CORE
1204
L2 CACHE CONTROL
1214
BUS INTERFACE UNIT
1216
L2 CACHE
1212
INTERCONNECT
1228
SPI MASTER
1224
FLASH
CONTROLLER
1222
SDRAM
CONTROLLER
1220
BOOT ROM
1218
SIM I/F
1234
FLASH
1232
DRAM
1230
POWER
CONTROL

FIGURE 12

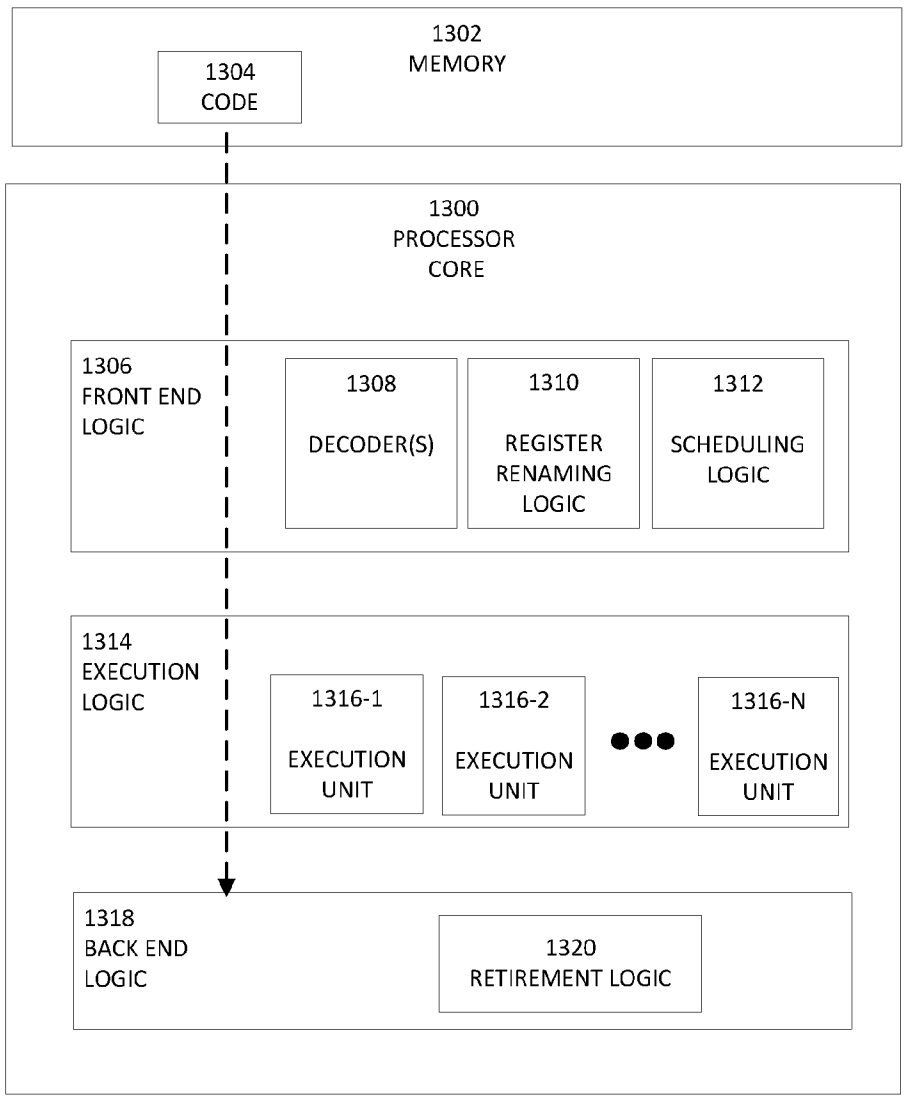

1302
MEMORY
1304
CODE
1300
PROCESSOR
CORE
1306
FRONT END
LOGIC
1308
DECODER(S)
1310
REGISTER
RENAMING
LOGIC
1312
SCHEDULING
LOGIC
1314
EXECUTION
LOGIC
1316-1
EXECUTION
UNIT
1316-2
EXECUTION
UNIT
1316-N
EXECUTION
UNIT
1318
BACK END
LOGIC
1320
RETIREMENT LOGIC

FIGURE 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2020/015283

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/33(2013.01)i, G06T 1/00(2006.01)i, H04L 9/32(2006.01)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)
G06F 21/33; G06F 11/30; G06F 21/00; G06F 21/56; G06K 9/00; H04L 12/26; G06T 1/00; H04L 9/32

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: image, screenshot, signature, exploration strategy, user interface elements

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012-0002839 A1 (JARNO NIEMELA et al.) 05 January 2012 paragraphs [0050]-[0071]; claims 1, 7; and figure 2	1-20
A	US 2017-0169224 A1 (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 15 June 2017 paragraphs [0024]-[0031]; claims 1, 3, 8; and figure 1	1-20
A	US 2013-0019310 A1 (YUVAL BEN-ITZHAK et al.) 17 January 2013 paragraphs [0023]-[0026]; claims 1, 3, 6; and figure 2	1-20
A	US 2013-0311653 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 21 November 2013 paragraphs [0105]-[0146]; claims 1, 6; and figures 7-11	1-20
A	US 9117078 B1 (HAO-LIANG CHIEN et al.) 25 August 2015 column 3, line 17 - column 4, line 19; claims 1, 7; and figure 1	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
"D" document cited by the applicant in the international application
"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
22 May 2020 (22.05.2020)	**22 May 2020 (22.05.2020)**

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
KIM, Sung Hee
Telephone No. +82-42-481-3516

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2020/015283

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0002839 A1	05/01/2012	US 8844039 B2	23/09/2014
US 2017-0169224 A1	15/06/2017	KR 10-2017-0068814 A	20/06/2017
		US 10339315 B2	02/07/2019
US 2013-0019310 A1	17/01/2013	US 2014-0331323 A1	06/11/2014
		US 2015-0113652 A1	23/04/2015
		US 8732831 B2	20/05/2014
		US 9288226 B2	15/03/2016
		US 9424422 B2	23/08/2016
US 2013-0311653 A1	21/11/2013	CN 103377109 A	30/10/2013
		US 2013-0275579 A1	17/10/2013
		US 9608881 B2	28/03/2017
		US 9628357 B2	18/04/2017
US 9117078 B1	25/08/2015	None	