



- (51) **International Patent Classification:**
G06F 8/61 (2018.01) *G06F 9/445* (2006.01)
- (21) **International Application Number:**
PCT/MY2019/050042
- (22) **International Filing Date:**
13 August 2019 (13.08.2019)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
PI 2018001447 15 August 2018 (15.08.2018) MY
- (71) **Applicant:** **MIMOS BERHAD** [MY/MY]; Technology Park Malaysia, Kuala Lumpur, 57000 (MY).
- (72) **Inventors:** **KANDAN, Rajendar**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **IS-**

MAIL, Bukhary Ikhwan; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **MOSTAJERAN GOORTANI, Ehsan**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **MOHD MYDIN, Mohd Nizam**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **KHALID, Mohammad Fairus**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(74) **Agent:** **PYPRUS SDN BHD**; Unit 1608, 16th Floor, Block A, Damansara Intan, NO. 1, Jalan SS20/27, Selangor, Petaling Jaya, 47400 (MY).

(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,

(54) **Title:** SYSTEM AND METHOD FOR IDENTIFYING CONTAINER IMAGE AND DYNAMICALLY CREATING THE CONTAINER IMAGE

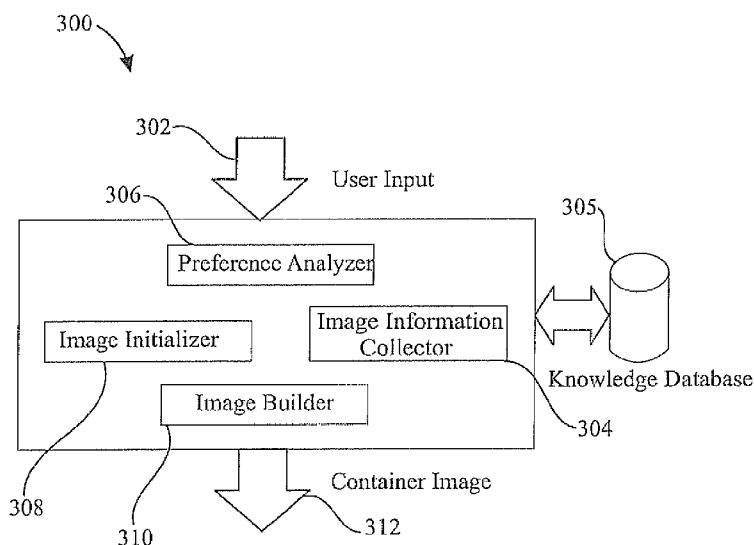


FIG. 3

(57) **Abstract:** Disclosed is a system and method for identifying a container image and dynamically creating the container image based on user preferences. The method includes a step of obtaining information about the container image from a plurality of sources to store in a database through an image information collector module (212). Then the method collects user preferences for analysis and prioritizing the user preferences through a preference analyzer module (214). The preference analyzer module (214) identifies the container image and initiates a negotiation on the user preferences. Further, the method prepares a virtual environment for creating the container image through an image initializer module (216). The image initializer module (216) initiates a downloading operation to download a plurality of software and further transfer the software to respective resources for creating the container image. Furthermore, the method includes the step of creating the container image based on the user preferences through an image builder module (218). The image builder module



KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

- *of inventorship (Rule 4.17(iv))*

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 IDENTIFYING CONTAINER IMAGE AND DYNAMICALLY CREATING THE CONTAINER IMAGE

FIELD OF INVENTION

5 [0001] The present invention relates to creating container image, in particular to a system and method for concerning a container image and dynamically creating the container image based on user preferences.

10 BACKGROUND

[0002] Typically, a software container automates and simplifies deployment of a software application over a cloud-computing platform or in a large enterprise network. A software container may comprise a standalone computing environment in which is installed one or more configured computer applications, infrastructure, and associated software. Further, the software container provides a layer of abstraction and virtualization for a running software application. A Docker® image is built up from a series of layers. Each layer of the Docker® image represents an instruction in the Dockerfile of the image.

20 [0003] US patent number 9,367,305 B1 filed by Yash Kumar et al. discloses a system for automatically generating containers and a relational communications graph for a source application comprises a processor to receive the source application. The source application includes a source element to determine a dependency component from the source element and generate a container configuration which includes the dependency component. Further, the processor executes the container configuration so that the operating system loads the dependency component within a container running under the operating system. However, the container configuration system disclosed in the Yash Kumar et al. reference is not able to expedite the process of building new container images by utilizing the existing container images.

30 [0004] US patent publication number US 2017/0300311 A1 filed by Daniel Vasquez Lopez et al. discloses a system for generating dynamic images. The system includes a processor to detect a request to generate a container image based on a policy file and identify

a host image from a host operating system. The processor further generates the container image based on the host image and the policy file. The policy file provides the first set of files to be copied from the host image to the container image. Further, a set of reparse points are created in the container image.

5

[0005] Another system to deploy dynamic container with parallel layers is disclosed in the US patent publication number US 2017/0277524 A1 filed by Badekila Ganesh Prashanth Bhat et al. The system includes a processor to create an image file by processing each instruction of an image-container file in sequential order. The processor at deployment
10 time determines that which layers to deploy as a function of a state of a condition using a method of parallel layers. Further, the parallel layers share a common stack level in the image file. When the image file deploys a container, only one parallel layer is selected for each level so that the deployed instance of the container includes one layer per level. However, the system disclosed in the Badekila Ganesh Prashanth Bhat et al. reference does not compare the
15 user requirements and preferences against the existing container images and also not able to expedite the process of building the new container images by utilizing the existing container images.

[0006] Accordingly, it is an objective of the present invention for a system and
20 method for addressing the aforesaid problems and setbacks.

SUMMARY

[0007] There is a need for an efficient and effective system and method for identifying a container image and dynamically creating the container image based on a plurality of user
25 preferences. Further, there is a need for a system and method that provides meta-data information to explain the container image. Furthermore, there is a need for a system and method to provide the technical information such as bugs, vulnerabilities, etc. about each layer of the container image.

[0008] The present invention mainly solves the technical problems existing in the prior
30 art. In response to these problems, the present invention provides a system and method for identifying a container image and dynamically creating the container image based on a plurality of user preferences.

5 [0009] An aspect of the present disclosure relates to a method for identifying a container image and dynamically creating the container image based on a plurality of user preferences. The method includes a step of obtaining information about the container image from a plurality of sources to store in a database through an image information collector module.

10 [0010] Then the method includes a step of collecting a plurality of user preferences for analysis and prioritizing the user preferences through a preference analyzer module. The preference analyzer module further identifies the container image and initiates a negotiation on the user preferences. Further, the method includes a step of preparing a virtual environment for creating the container image through an image initializer module. The image initializer module initiates a downloading operation to download a plurality of software and further transfer the plurality of software to a plurality of respective resources for creating the container image. The method then includes a step of creating the container image based on the user preferences through an image builder module. The image builder module further scans the created container image to provide technical information about the container image.

[0011] In an aspect, the step of collecting user preferences for analysis and prioritizing the user preferences further comprises a step of verifying the collected user preferences against a plurality of rules to identify one or more invalid user preference.

20 [0012] In an aspect, the identified invalid user preference displayed over a display module.

[0013] In an aspect, the plurality of user preferences are selected from at least one of an operating system, version of a software, size of the software, recommended release of the software, a plurality of grade level of the software, and a combination thereof.

25 [0014] In an aspect, the obtaining information about the container image comprises operating system data, application data, type of build data, star rating data, number of download data, and vulnerable data.

[0015] In an aspect, the step of prioritizing the user preferences further comprises a step of generating a search identification number to check historical preferences of a user.

[0016] In an aspect, the step of identifying the container image and initiating a negotiation on the user preferences further comprises a step of identifying a suitable index of the container image based on the collected user preferences. Then this step includes the step of negotiating on the user preferences to select an appropriate container image, and further acknowledges and provides information about the container image based on the negotiation on the user preferences.

[0017] In an aspect, the step of preparing the virtual environment for creating the container image further comprises a step of preparing a plurality of build sources and storing a template identification number in the database.

[0018] An aspect of the present disclosure relates to a device in a network. The device includes a non-transitory storage device having embodied therein one or more routines operable to identify a container image and dynamically creating the container image based on a plurality of user preferences. The one or more routines include an image information collector module, a preference analyzer module, an image initializer module, and an image builder module.

[0019] The image information collector module is configured to obtain information about the container image from a plurality of sources to store in a database. The preference analyzer module is configured to collect a plurality of user preferences for analysis and prioritizing the user preferences. The preference analyzer module further identifies the container image and initiates a negotiation on the user preferences. The image initializer module is configured to prepare a virtual environment for creating the container image. The image initializer module initiates a downloading operation to download a plurality of software and further transfer the plurality of software to a plurality of respective resources for creating the container image. The image builder module is configured to create the container image based on the user preferences. The image builder module further scans the created container image to provide technical information about the container image.

[0020] An aspect of the present disclosure relates to a system to identify a container image and dynamically create the container image based on a plurality of user preferences. The system includes a processor and a memory. The memory stores machine-readable instructions that when executed by the processor cause the processor to obtain information about the container image from a plurality of sources to store in a

database through an image information collector module. Then the processor is configured to collect a plurality of user preferences for analysis and prioritizing the user preferences through a preference analyzer module. The preference analyzer module further identifies the container image and initiates a negotiation on the user preferences.

5 The processor is then configured to prepare a virtual environment for creating the container image through an image initializer module. The image initializer module initiates a downloading operation to download a plurality of software and further transfer the plurality of software to a plurality of respective resources for creating the container image. The processor is further configured to create the container image based on the user

10 preferences through an image builder module. The image builder module further scans the created container image to provide technical information about the container image.

[0021] Accordingly, one advantage of the present invention is that it enables a user to view the technical information about the related container images.

[0022] Accordingly, one advantage of the present invention is that it enables the user to

15 select the best and appropriate suitable container image for creation based on their specific preferences.

[0023] Accordingly, one advantage of the present invention is that it enables the user dynamically create the container images.

[0024] Other features of embodiments of the present disclosure will be apparent from

20 accompanying drawings and from the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] In the figures, similar components and/or features may have the same reference

25 label. Further, various components of the same type may be distinguished by following the reference label with a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description applies to any one of the similar components having the same first reference label irrespective of the second reference label.

[0026] FIG. 1 illustrates a network implementation of the container image system and method to identify a container image and dynamically create the container image based on a plurality of user preferences, in accordance with an embodiment of the present subject matter.

5 [0027] FIG. 2 illustrates the container image system to identify a container image and dynamically create the container image based on a plurality of user preferences, in accordance with an embodiment of the present subject matter.

10 [0028] FIG. 3 illustrates the architecture of the container image system, in accordance with an embodiment of the present subject matter.

[0029] FIG. 4 illustrates a flowchart of the method for identifying a container image and dynamically creating the container image based on a plurality of user preferences, in accordance with at least one embodiment.

15

[0030] FIG. 5 illustrates a flowchart of the operations involves in verifying the collected user preferences against a plurality of rules to identify one or more invalid user preference through a preference analyzer module, in accordance with at least one embodiment.

20

[0031] FIG. 6 illustrates a flowchart of the operations involves in prioritizing the user preferences through the preference analyzer module, in accordance with at least one embodiment.

25 [0032] FIG. 7 illustrates a flowchart of the operations involves in identifying the container image and initiating a negotiation on the user preferences through a preference analyzer module, in accordance with at least one embodiment.

30 [0033] FIG. 8 illustrates a flowchart of the operations involves in preparing a virtual environment for creating the container image through an image initializer module, in accordance with at least one embodiment.

[0034] FIG. 9 illustrates a flowchart of the operations involves in creating the container image based on the user preferences through an image builder module, in accordance with at least one embodiment.

5

DETAILED DESCRIPTION

[0035] Systems and methods are disclosed for identifying a container image and dynamically creating the container image based on a plurality of user preferences. Embodiments of the present disclosure include various steps, which will be described below. The steps may be performed by hardware components or may be embodied in machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor programmed with the instructions to perform the steps. Alternatively, steps may be performed by a combination of hardware, software, firmware, and/or by human operators.

15

[0036] Embodiments of the present disclosure may be provided as a computer program product, which may include a machine-readable storage medium tangibly embodying thereon instructions, which may be used to program a computer (or other electronic devices) to perform a process. The machine-readable medium may include, but is not limited to, fixed (hard) drives, magnetic tape, floppy diskettes, optical disks, compact disc read-only memories (CD-ROMs), and magneto-optical disks, semiconductor memories, such as ROMs, PROMs, random access memories (RAMs), programmable read-only memories (PROMs), erasable PROMs (EPROMs), electrically erasable PROMs (EEPROMs), flash memory, magnetic or optical cards, or other type of media/machine-readable medium suitable for storing electronic instructions (e.g., computer programming code, such as software or firmware).

[0037] Various methods described herein may be practiced by combining one or more machine-readable storage media containing the code according to the present disclosure with appropriate standard computer hardware to execute the code contained therein. An apparatus for practicing various embodiments of the present disclosure may involve one or more computers (or one or more processors within a single computer) and storage systems containing or having network access to computer program(s) coded in accordance with

30

various methods described herein, and the method steps of the disclosure could be accomplished by modules, routines, subroutines, or subparts of a computer program product.

[0038] The present invention discloses a system and method whereby the container
5 images are available online for the users. The user are able find the corresponding meta-data
information which explains the container image or the technical information such as bugs,
vulnerabilities about each layer of the container image is readily accessible thereby. The
system and method is adapted to generate the container image based on the policy file and
and it offers mechanisms to dynamically build the desired container image based on the user
10 preferences.

[0039] Although the present disclosure has been described with the purpose of
identifying a container image and dynamically creating the container image based on a
plurality of user preferences, it should be appreciated that the same has been done merely to
15 illustrate the invention in an exemplary manner and any other purpose or function for which
explained structures or configurations could be used, is covered within the scope of the
present disclosure.

[0040] Exemplary embodiments will now be described more fully hereinafter with
20 reference to the accompanying drawings, in which exemplary embodiments are shown. This
invention may, however, be embodied in many different forms and should not be construed
as limited to the embodiments set forth herein. These embodiments are provided so that this
disclosure will be thorough and complete and will fully convey the scope of the invention to
those of ordinary skill in the art. Moreover, all statements herein reciting embodiments of the
25 invention, as well as specific examples thereof, are intended to encompass both structural and
functional equivalents thereof. Additionally, it is intended that such equivalents include both
currently known equivalents as well as equivalents developed in the future (i.e., any elements
developed that perform the same function, regardless of structure).

30 **[0041]** Thus, for example, it will be appreciated by those of ordinary skill in the art that
the diagrams, schematics, illustrations, and the like represent conceptual views or processes
illustrating systems and methods embodying this invention. The functions of the various
elements shown in the figures may be provided through the use of dedicated hardware as well

as hardware capable of executing associated software. Similarly, any switches shown in the figures are conceptual only. Their function may be carried out through the operation of program logic, through dedicated logic, through the interaction of program control and dedicated logic, or even manually, the particular technique being selectable by the entity
5 implementing this invention. Those of ordinary skill in the art further understand that the exemplary hardware, software, processes, methods, and/or operating systems described herein are for illustrative purposes and, thus, are not intended to be limited to any particular name.

10 **[0042]** Specific details are given in the following description to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these specific details. For example, circuits, systems, networks, processes, and other components may be shown as components in block diagram form in order not to obscure the embodiments in unnecessary detail. In
15 other instances, well-known circuits, processes, algorithms, structures, and techniques may be shown without unnecessary detail to avoid obscuring the embodiments.

[0043] The term “machine-readable storage medium” or “computer-readable storage medium” includes, but is not limited to, portable or non-portable storage devices, optical
20 storage devices, and various other mediums capable of storing, containing, or carrying instruction(s) and/or data. A machine-readable medium may include a non-transitory medium in which data can be stored, and that does not include carrier waves and/or transitory electronic signals propagating wirelessly or over wired connections. Examples of a non-transitory medium may include but are not limited to, a magnetic disk or tape, optical storage
25 media such as compact disk (CD) or versatile digital disk (DVD), flash memory, memory or memory devices.

[0044] **FIG. 1** illustrates a network implementation **100** an container image system **102** and method to identify a container image and dynamically create the container image based
30 on a plurality of user preferences, in accordance with an embodiment of the present invention. Although the present subject matter is explained considering that the container image system **102** is implemented on a server, it may be understood that the container image system **102** may also be implemented in a variety of computing systems, such as a laptop

computer, a desktop computer, a notebook, a workstation, a mainframe computer, a server, a network server, and the like. It will be understood that the container image system **102** may be accessed by multiple users through one or more computing devices **104-1**, **104-2**...**104-N**, collectively referred to as computing unit **104** hereinafter, or applications residing on the computing unit **104**. Examples of the computing unit **104** may include but are not limited to, a portable computer, a personal digital assistant, a handheld or mobile device, smart devices, and a workstation. The computing units **104** are communicatively accessible to the container image system **102** through a network **106**.

[0045] In one implementation, the network **106** may be a wireless network, a wired network or a combination thereof. The network **106** can be implemented as one of the different types of networks, such as an intranet, local area network (LAN), wide area network (WAN), the internet, and the like. The network **106** may either be a dedicated network or a shared network. The shared network represents an association of the different types of networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Wireless Application Protocol (WAP), and the like, to communicate with one another. Further, the network **106** may include a variety of network devices, including routers, bridges, servers, computing devices, storage devices, and the like.

20

[0046] **FIG. 2** illustrates the container image system **102** of **FIG. 1** to identify a container image and dynamically create the container image based on a plurality of user preferences, in accordance with an embodiment of the present invention. The system **102** may include at least one processor **202**, an input/output (I/O) interface **204**, and a memory **206**.

25

[0047] The processor **202** may be implemented as one or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, and/or any devices that manipulate signals based on operational instructions. Among other capabilities, the at least one processor **202** is configured to fetch and execute computer-readable instructions stored in the memory **206**.

30

[0048] The I/O interface **204** may include a variety of software and hardware interfaces, for example, a web interface, a graphical user interface, and the like. The I/O interface **204** may allow the system **102** to interact with a user directly or through the computing unit **104**. Further, the I/O interface **204** may enable the system **102** to communicate with other computing devices, such as web servers and external data servers (not shown). The I/O interface **204** can facilitate multiple communications within a wide variety of networks and protocol types, including wired networks, for example, LAN, cable, etc., and wireless networks, such as WLAN, cellular, or satellite. The I/O interface **204** may include one or more ports for connecting a number of devices to one another or to another server.

[0049] The memory **206** may include any computer-readable medium known in the art including, for example, volatile memory, such as static random access memory (SRAM) and dynamic random access memory (DRAM), and/or non-volatile memory, such as read-only memory (ROM), erasable programmable ROM, flash memories, hard disks, optical disks, and magnetic tapes. The memory **206** may include modules **208** and data **210**.

[0050] The modules **208** include routines, programs, objects, components, data structures, etc., which perform particular tasks or implement particular abstract data types. In one implementation, the modules **208** may include an image information collector module **212**, a preference analyzer module **214**, an image initializer module **216**, an image builder module **218**, and other module **222**. The other modules **222** may include programs or coded instructions that supplement applications and functions of the system **102**.

[0051] The data **210**, amongst other things, serves as a repository for storing data processed, received, and generated by one or more of the modules **208**. The data **210** may also include an image information collector data **223**, a preference analyzer data **224**, an image initializer data **225**, an image builder data **226** and other data **234**. The other data **234** may include data generated as a result of the execution of one or more modules in the other module **222**.

[0052] In one implementation, the image information collector module **212** is configured to obtain information about the container image from a plurality of sources to

store in a database. Examples of the plurality of sources include but are not no limited to <https://hub.docker.com/explore/>, <https://bitnami.com/containers#turnkey-containers>, or any platform utilized by developers to build, package, maintain and run distributed applications. The preference analyzer module **214** is configured to collect a plurality of user preferences for analysis and prioritizing the user preferences. For example, the user prefers to use a specific version of the operating system (OS) such as “Ubuntu 16.04”, or “Ubuntu 18.04”. Similarly, the user prefers to use a specific version of the application. The preference analyzer module **214** accordingly collects the preferred OS version and application version for analysis. In an embodiment, the preference analyzer module **214** further generates a search identification number to check historical preferences of a user.

[0053] In an embodiment, the plurality of user preferences are selected from at least one of an operating system, version of a software, size of the software, recommended release of the software, a plurality of grade level of the software, and a combination thereof. Typically, the operation of the software versions depends on the version of the operating system in use. Thus the top priority for the preference analyzer module **214** is to select the version of the operating system to find the appropriate software version. For example, the default Docker version 1.11 is available in Ubuntu OS 14.04 (trusty) and a default Docker version 17.05 is available in Ubuntu OS 16.04. In an embodiment, the obtained information about the container image comprises operating system data, application data, type of build data, star rating data, number of download data, and vulnerable data.

[0054] The preference analyzer module **214** further identifies the container image and initiates a negotiation on the user preferences. In an embodiment, the preference analyzer module **214** further verifies the collected user preferences against a plurality of rules to identify one or more invalid user preference. In case the required software A is not available in the OS version X, then predefined rules eliminate the failure condition by having a list of software versions available to that respective OS. The present invention utilizes a knowledge repository for references to identify the software list and available versions. The identified invalid user preferences are displayed over a display module. In an embodiment, the preference analyzer module **214** further identifies a suitable index of the container image based on the collected user preferences; negotiates on the user preferences to select an

appropriate container image, and acknowledges and provides information about the container image based on the negotiation on the user preferences.

[0055] The image initializer module **216** is configured to prepare a virtual environment for creating the container image. The image initializer module **216** initiates a downloading operation to download a plurality of software and further transfer the plurality of software to a plurality of respective resources. In an exemplary embodiment, the image initializer module **216** enables a user to download the software by using the command line. For example, docker pull Ubuntu:16.04 <ImageName>, wget <url> etc. In other embodiment, a Graphical User Interface (GUI) may be adapted.

[0056] In an embodiment, the resources are the servers comprises adequate memory and storage space for managing the creation of the container images. Based on the requirement, the resources are identified in run-time for building the container images. In an additional embodiment, the image initializer module **216** manages the download of the container image and the pre-requisite. The image builder module **218** is configured to create the container image based on the user preferences. The image builder module **218** further scans the created container image to provide technical information about the container image. In an embodiment, the image builder module **218** prepares a plurality of build sources and stores a template identification number in the database. Table 1 below exemplifies some technical information of the container image built. The build sources are used when the container image is required to build with the user-defined files (e.g., scripts, and files). Examples of the build include but are not limited to command: docker builds -t ubuntu: user -f Dockerfile /imagesource-directory. This exemplary command builds the images with all the files listed in the image source directory which contains all the files uploaded by the user. In an embodiment, the template identification number is auto-generated and labeled. For example, "newbuild10": Description: Tomcat8+ubuntu16.04, which can be referenced, if a similar request made from the other user to build the container images in less time.

[0057] FIG. 3 illustrates the architecture **300** of the system in accordance with an embodiment of the present subject matter. The image information collector **304** obtains the information from the sources and update in knowledge database **305**. Thus all detailed information about the images are available in the knowledge database **305**. Table 1 illustrates

the obtained container image information. The preference analyzer **306** collects the user preferences **302** and negotiates with the desired requirement. The user inputs are verified and process to the image initializer **308** for further processing. Then the invalid inputs are identified and displayed to the user. The image initializer **308** prepares the environment for building the container image. All necessary software are downloaded and moved or copied to the respective resource for building. The image builder **310** creates the desired image based on the user preferences. Further, the image builder **310** scans the built image and provide the summary of the image details. Thus the container image system and method collect the user preferences and negotiate for proper selection of the image. Further, the container image system and method prepare a suitable environment for building the image based on the comparative analysis of the existing image and build a new image based on the user preferences.

Table 1: Image built information

User preferences	Existing image	Image built	Built report
Ubuntu latest	Ubuntu:14.04	Ubuntu : 16.04	State : Successful
Tomcat latest	Tomcat 8	Tomcat 8.5	Scan results :
Size < 500 MB	379 MB	425 MB	(common vulnerable scoring system) CVSS_Score : 5.08 Total_Cve (Common vulnerabilities and exposure) : 52 CVE_severity (total : 52 - High : 7, Medium : 45)

[0058] **FIG. 4** illustrates a flowchart **400** of the method for identifying a container image and dynamically creating the container image based on a plurality of user preferences, in accordance with at least one embodiment. The steps of **FIG. 4** are explained in conjunction with FIGs. 5-9. The method initiates with a step **402** of obtaining information about the container image from a plurality of sources to store in a database through an image

information collector module 212. In an embodiment, the obtained information pertaining to the container image comprises operating system data, application data, type of build data, star rating data, number of download data, and vulnerable data.

5 **[0059]** Then the method includes a step **404** of collecting a plurality of user preferences for analysis and prioritizing the user preferences through a preference analyzer module 214. The preference analyzer module 214 further identifies the container image and initiates a negotiation on the user preferences. In an embodiment, the plurality of user preferences are selected from at least one of an operating system, version of a software, size of the software,
10 recommended release of the software, a plurality of grade level of the software, and/or a combination thereof.

[0060] In step **406**, a virtual environment is created for creating the container image. The virtual environment is created through the image initializer module **216**. In step **408**, a
15 container image is created based on the user preference. The container image is created by the image builder module **218**. More detailed operations of steps **406** and **408** will be provided later below along with **FIG. 8** and **9** respectively.

[0061] **FIG. 5** illustrates a flowchart **500** of the operations involve in verifying the
20 collected user preferences against a plurality of rules to identify one or more invalid user preference through a preference analyzer module, in accordance with at least one embodiment of the present invention. The step **404** of collecting user preferences for analysis and prioritizing the user preferences further comprises a step **504** of verifying the collected user preferences against a plurality of rules such as the size of the software applications and
25 version conflicts between the various software applications to identify one or more invalid user preference. In case, the verification of the initial user preferences or user's input fails then the invalid user preference is displayed **502** over a display module. Further, if the collected user inputs are verified to be valid, the user is prompt for inputs for customization **506**. If the user needs customization, the user input data is stored **508** and then the system
30 stores the category details **510** of the user input data. In case the user is not willing to customize the user input data then the system directly stores category details **510** of the user input data. Examples of the category details include but not limited to whether the request is for a new build of the container image or the system has to initiate the new search. In case

the similar container images are not available, the system has to build a new container image from scratch or update the existing container image with the required software received from the user.

5 **[0062]** **FIG. 6** illustrates a flowchart **600** of the operations involve in prioritizing the user preferences through the preference analyzer module 214, in accordance with one embodiment of the present invention. The step of prioritizing the user preferences comprises a step **602** of generating a search identification number to check **604** historical preferences of a user. If the historical preferences of the user are present, the system identifies **606** the
10 matching search identification number corresponding to the generated search identification number. In case the historical preferences of the user are not present, the system initiates order **608** for the historical preferences of the user. As soon as the system receives the historical preferences of the user, the system updates **610** the search identification number.

15 **[0063]** **FIG. 7** illustrates a flowchart **700** of the operations involve in identifying the container image and initiating a negotiation on the user preferences through a preference analyzer module 214, in accordance with one embodiment of the present invention. The step of identifying the container image and initiating a negotiation on the user preferences further comprises a step **702** of identifying a suitable index of the container image based on the
20 collected user preferences. Then the method includes the step **704** of negotiating on the user preferences to select an appropriate container image. After that, the method includes the step **706** of acknowledging and providing information about the container image based on the negotiation on the user preferences.

25 **[0064]** **FIG. 8** illustrates a flowchart **800** of the operations involve in preparing a virtual environment for creating the container image through an image initializer module 216, in accordance with one embodiment of the present invention. The method includes a step **406** of preparing a virtual environment for creating the container image through an image initializer module 216. For a new creation or build **802** of the container image, the image
30 initializer module 216 initiates a downloading operation to download a plurality of software and further transfer or copy **804** the plurality of software to a plurality of respective resources for creating the container image. Then the method initiates order **806** about the dependencies of the software. The step **406** of preparing the virtual environment for creating the container

image further comprises a step **808** of preparing a plurality of build sources, and a step **810** of updating or storing a template identification number in the database.

[0065] **FIG. 9** illustrates a flowchart **900** of the operations involve in creating the container image based on the user preferences through an image builder module 218, in accordance with one embodiment of the present invention. The method includes a step **408** of creating the container image based on the user preferences through an image builder module 218. Then the step **408** of creating the container image further comprises a step **902** of initiating a process to create the container image. The image builder module further scans **904** the created container image to display **906** technical information about the container image.

[0066] Thus the container image system and method provide an efficient, simpler and more elegant framework for identifying a container image and dynamically creating the container image based on a plurality of user preferences. Further, the container image system and method provide meta-data information to explain the container image. Additionally, the container image system and method provide the technical information such as bugs, vulnerabilities, etc. about each layer of the container image.

[0067] While embodiments of the present disclosure have been illustrated and described, it will be clear that the disclosure is not limited to these embodiments only. Numerous modifications, changes, variations, substitutions, and equivalents will be apparent to those skilled in the art, without departing from scope of the disclosure, as described in the claims.

CLAIMS:

1. A method implemented by one or more processors (202) for identifying a container image and dynamically creating the container image based on a plurality of user preferences, characterized in that the method comprising:

obtaining, by one or more processors (202), information pertaining to the container image from a plurality of sources to store in a database through an image information collector module (212);

collecting, by one or more processors (202), a plurality of user preferences for analyzing and prioritizing the user preferences through a preference analyzer module (214), wherein the preference analyzer module (214) further identifies the container image and initiates a negotiation on the user preferences, wherein the plurality of user preferences are selected from at least one of an operating system, version of a software, size of the software, recommended release of the software, a plurality of grade level of the software, and/or a combination thereof;

preparing, by one or more processors (202), a virtual environment for creating the container image through an image initializer module (216), wherein the image initializer module (216) initiates a downloading operation to download a plurality of software and further transfer the plurality of software to a plurality of respective resources for creating the container image; and

creating, by one or more processors (202), the container image based on the user preferences through an image builder module (218), wherein the image builder module (218) further scans the created container image to provide technical information pertaining to the container image.

2. The method according to claim 1, wherein the step of collecting user preferences for analyzing and prioritizing the user preferences further comprises a step of verifying, by one or more processors (202), the collected user preferences against a plurality of rules to identify one or more invalid user preference, wherein the identified invalid user preference displayed over a display module.

3. The method according to claim 1, wherein the obtained information pertaining to the container image comprises operating system data, application data, type of build data, star rating data, number of download data, and vulnerable data.
- 5 4. The method according to claim 1, wherein the step of prioritizing the user preferences further comprises a step of generating, by one or more processors (202), a search identification number to check historical preferences of a user.
- 10 5. The method according to claim 1, wherein the step of identifying the container image and initiating a negotiation on the user preferences further comprises a step of:
identifying, by one or more processors (202), a suitable index of the container image based on the collected user preferences;
negotiating, by one or more processors (202), on the user preferences to select an appropriate container image; and
15 acknowledging and providing, by one or more processors (202), information pertaining to the container image based on the negotiation on the user preferences.
- 20 6. The method according to claim 1, wherein the step of preparing the virtual environment for creating the container image further comprises a step of preparing, by one or more processors (202), a plurality of build sources, and storing a template identification number in the database.
- 25 7. A system to identify a container image and dynamically create the container image based on a plurality of user preferences, characterized in that, the system comprising:
a memory (206) to store machine readable instructions pertaining to identification and creation of the container image; and
a processor (202) coupled to the memory and operable to execute the machine readable instructions stored in the memory, wherein the processor (202) comprises:
30 an image information collector module (212) to obtain information pertaining to the container image from a plurality of sources to store in a database, wherein the obtained information pertaining to the container image comprises operating system data, application data, type of build data, star rating data, number of download data, and vulnerable data;

- 5 a preference analyzer module (214) to collect a plurality of user preferences for analyzing and prioritizing the user preferences, wherein the preference analyzer module (214) further identifies the container image and initiates a negotiation on the user preferences to select an appropriate container image, wherein the plurality of user preferences are selected from at least one of an operating system, version of a software, size of the software, recommended release of the software, a plurality of grade level of the software, and/or a combination thereof;
- 10 an image initializer module (216) to prepare a virtual environment to create the container image, wherein the image initializer module (216) initiates a downloading operation to download a plurality of software and transfers the plurality of software to a plurality of respective resources to create the container image; and
- 15 an image builder module (218) to create the container image based on the user preferences, wherein the image builder module (218) further scans the created container image to provide technical information pertaining to the container image.
- 20 8. The system according to claim 7, wherein the preference analyzer module (214) verifies the collected user preferences against a plurality of rules to identify one or more invalid user preference; and acknowledges and provides information pertaining to the container image based on the negotiation on the user preferences.
- 25 9. The system according to claim 7, wherein the preference analyzer module (214) generates a search identification number to check historical preferences of a user; and identifies a suitable index of the container image based on the collected user preferences.
- 30 10. The system according to claim 7, wherein the image initializer module (216) prepares a plurality of build sources; and stores a template identification number in the database.

1/9

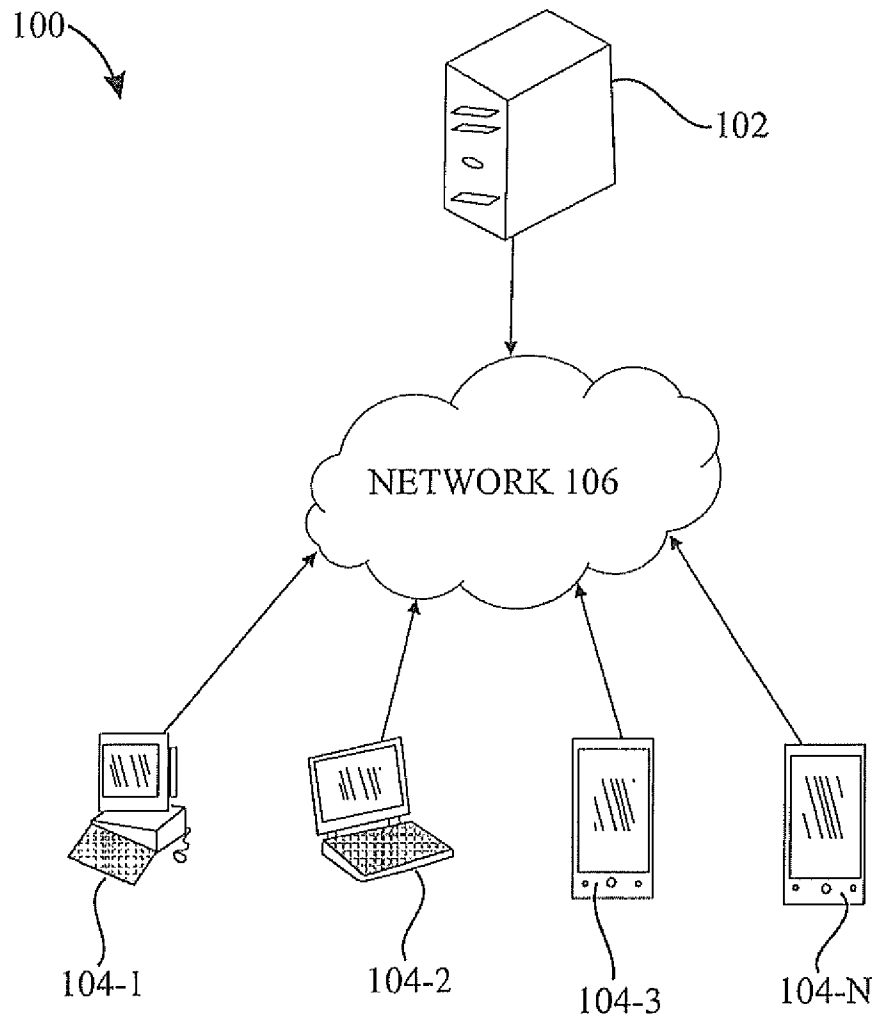


FIG. 1

2/9

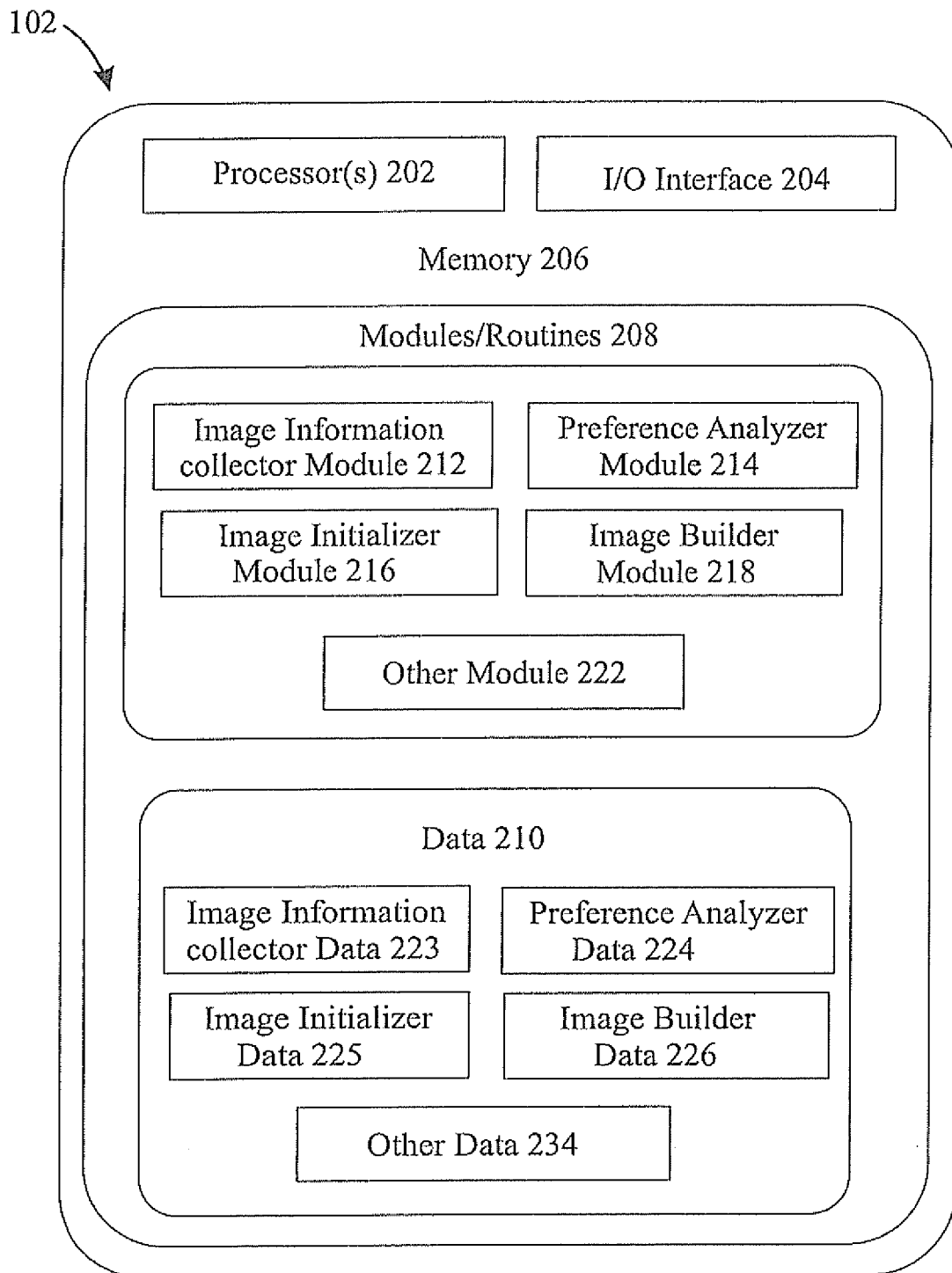


FIG. 2

3/9

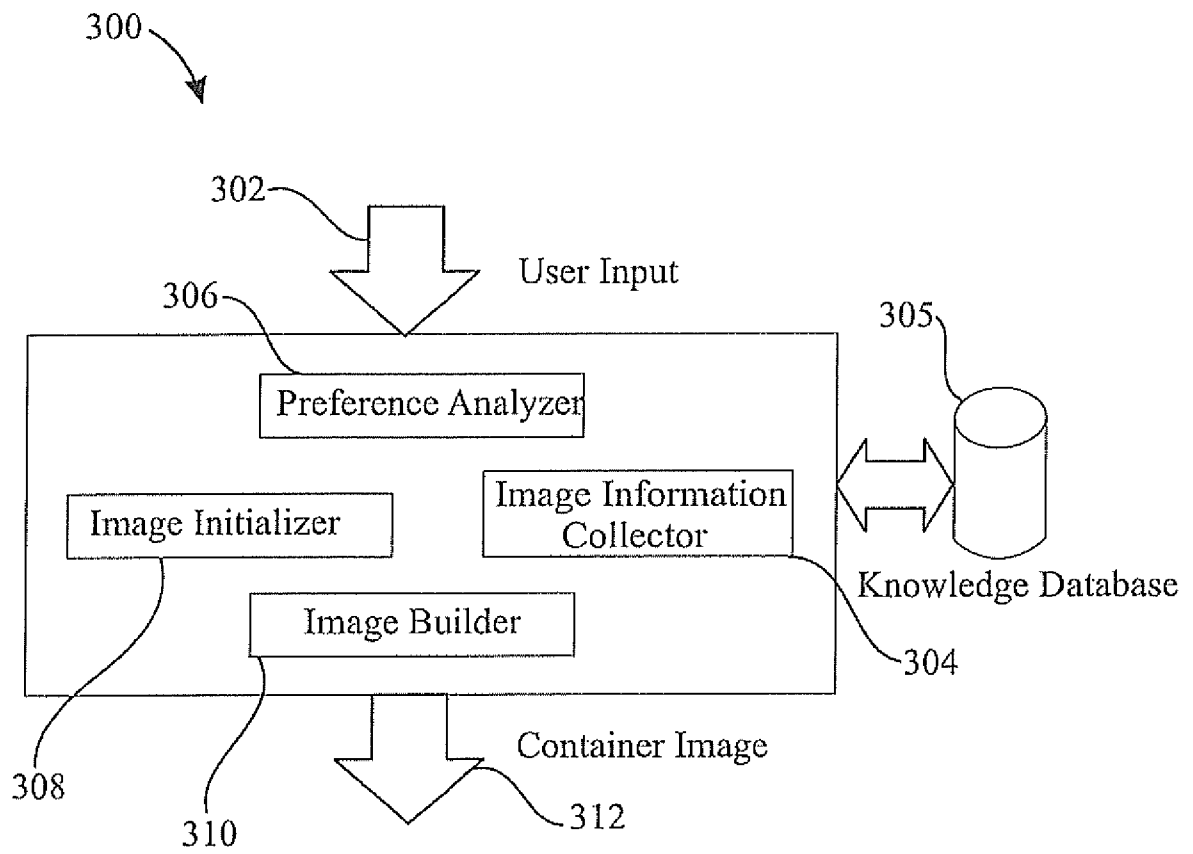


FIG. 3

4/9

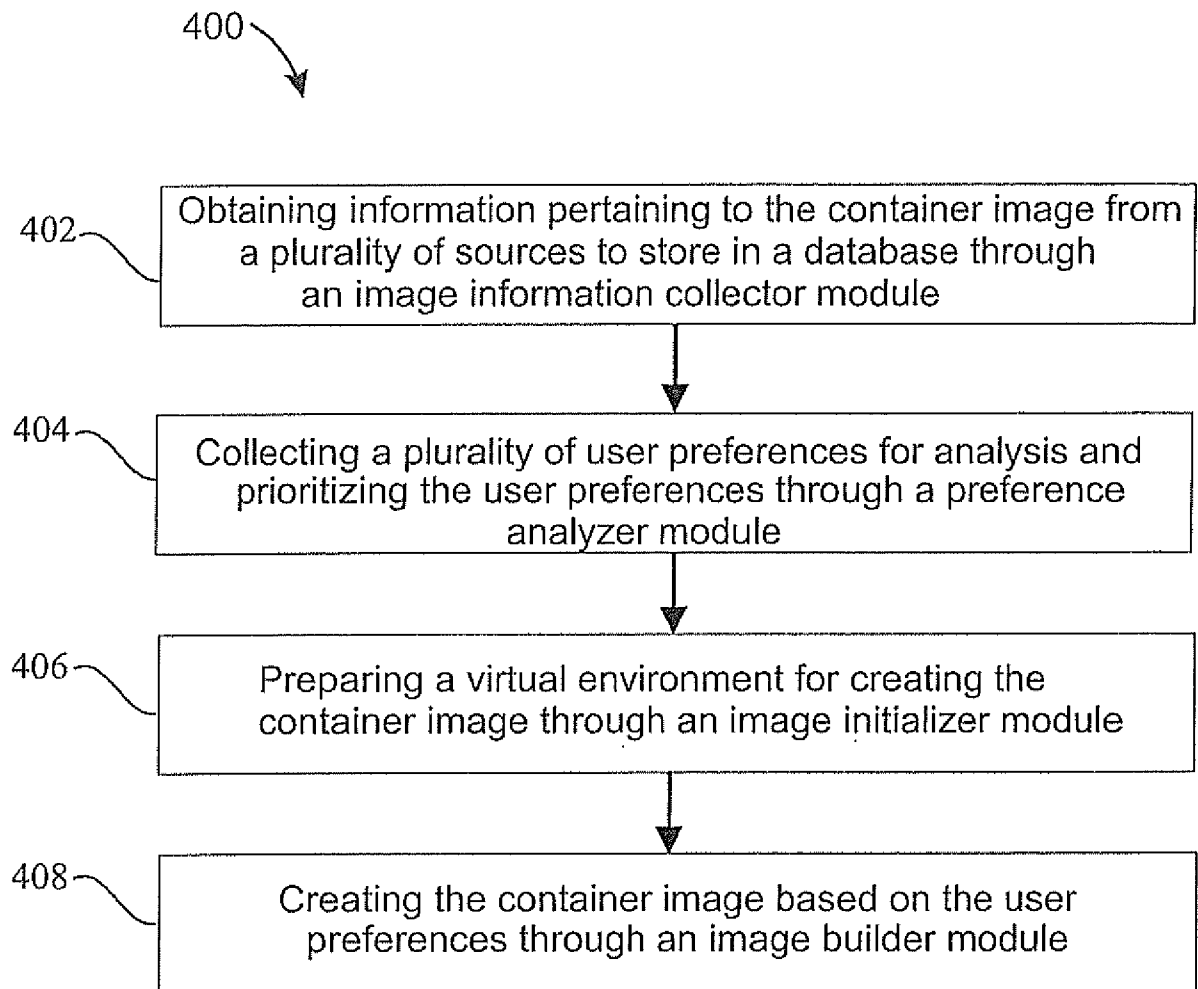


FIG. 4

5/9

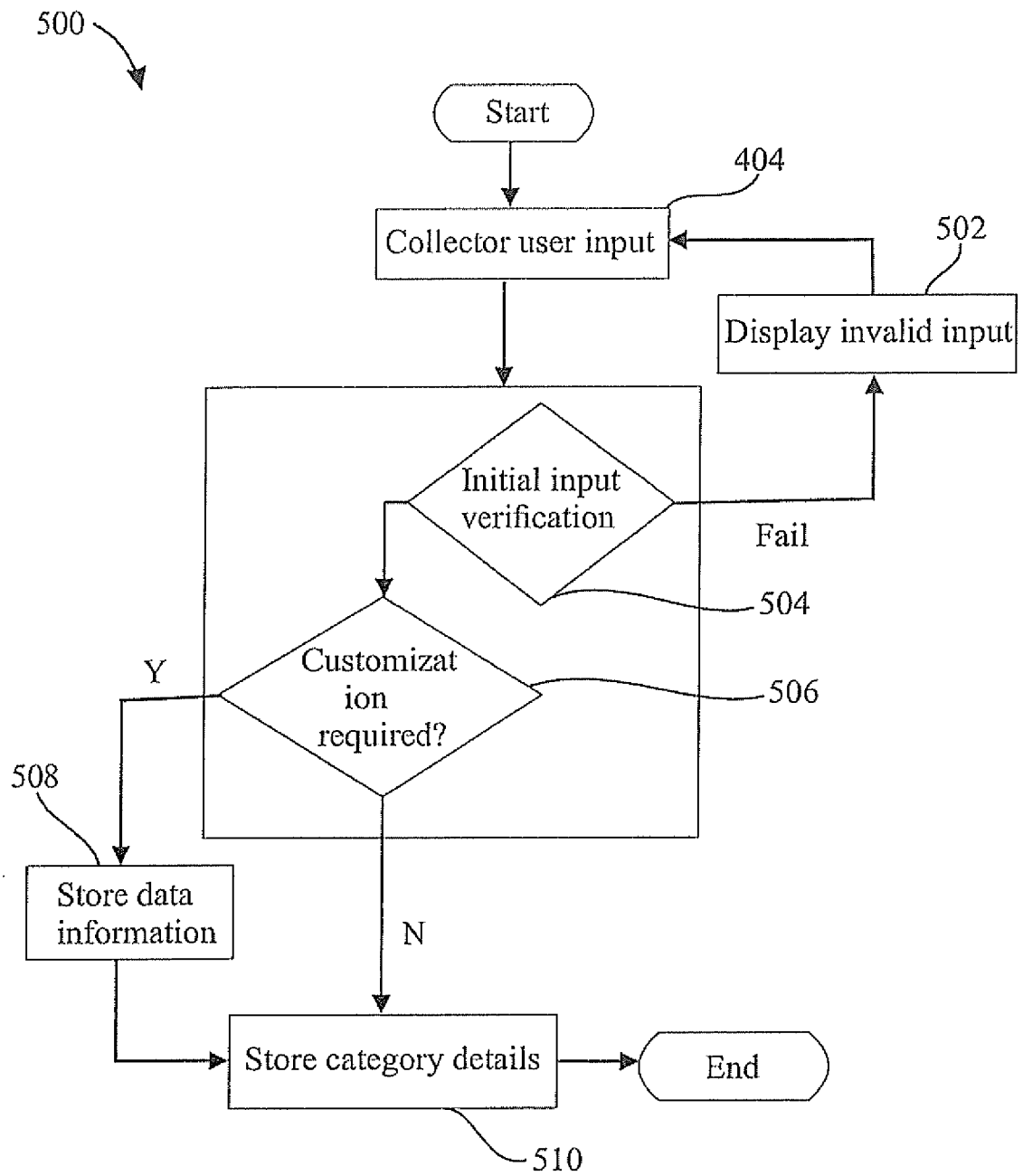


FIG. 5

6/9

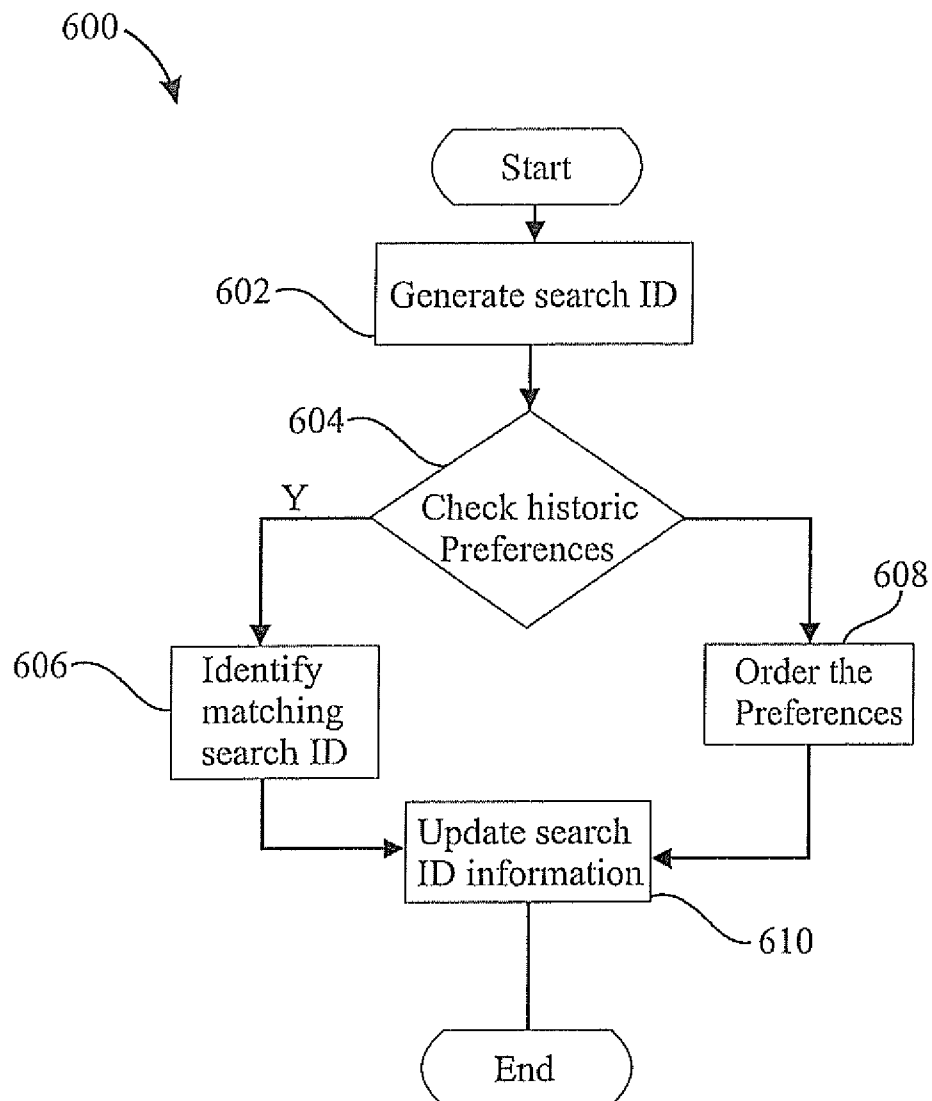


FIG. 6

7/9

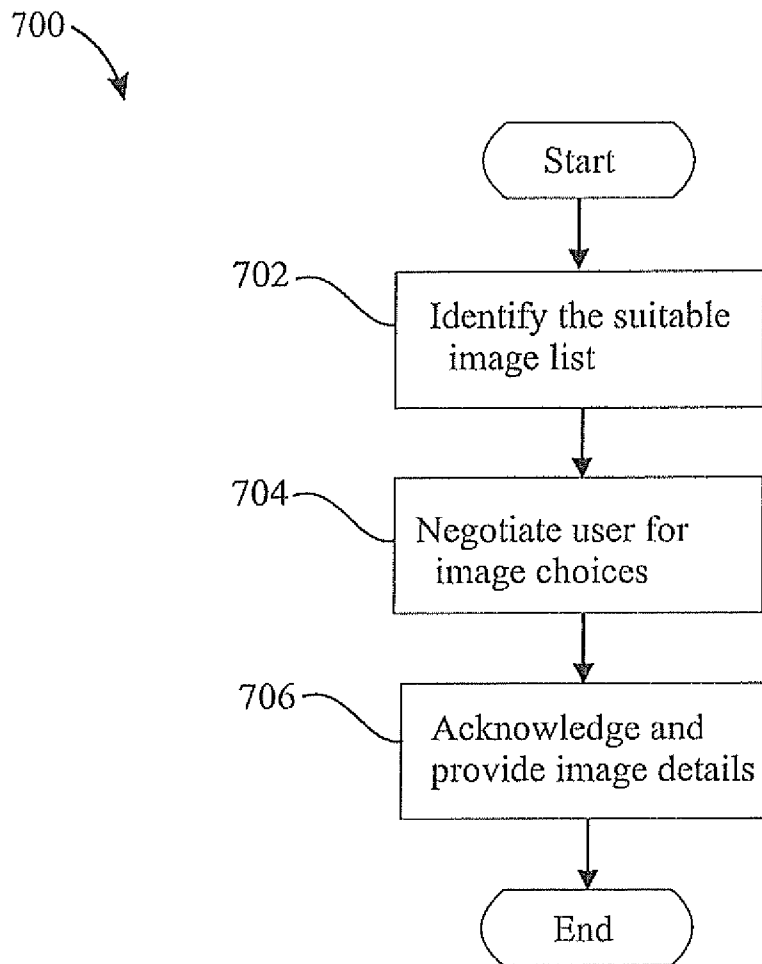


FIG. 7

8/9

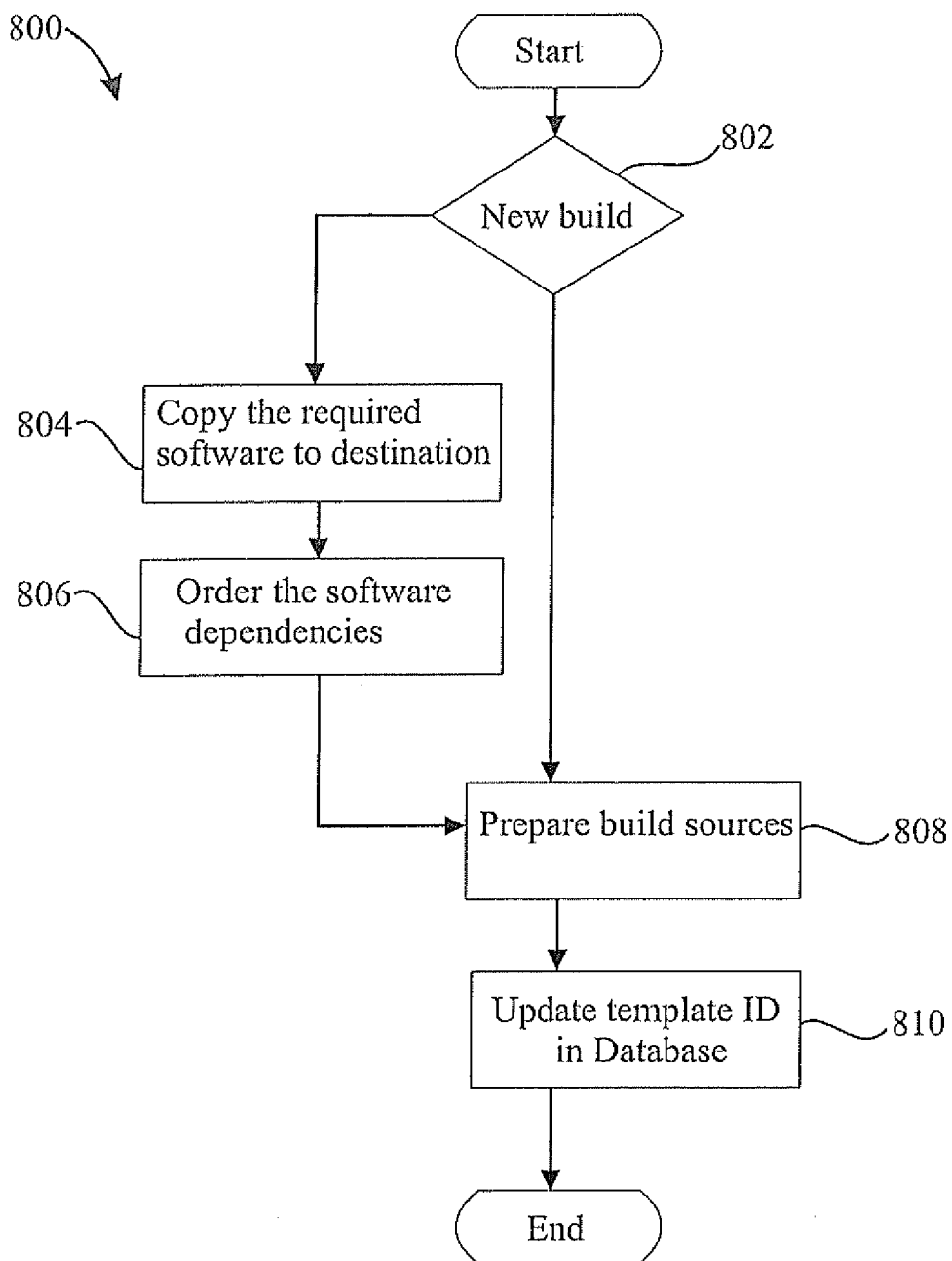


FIG. 8

9/9

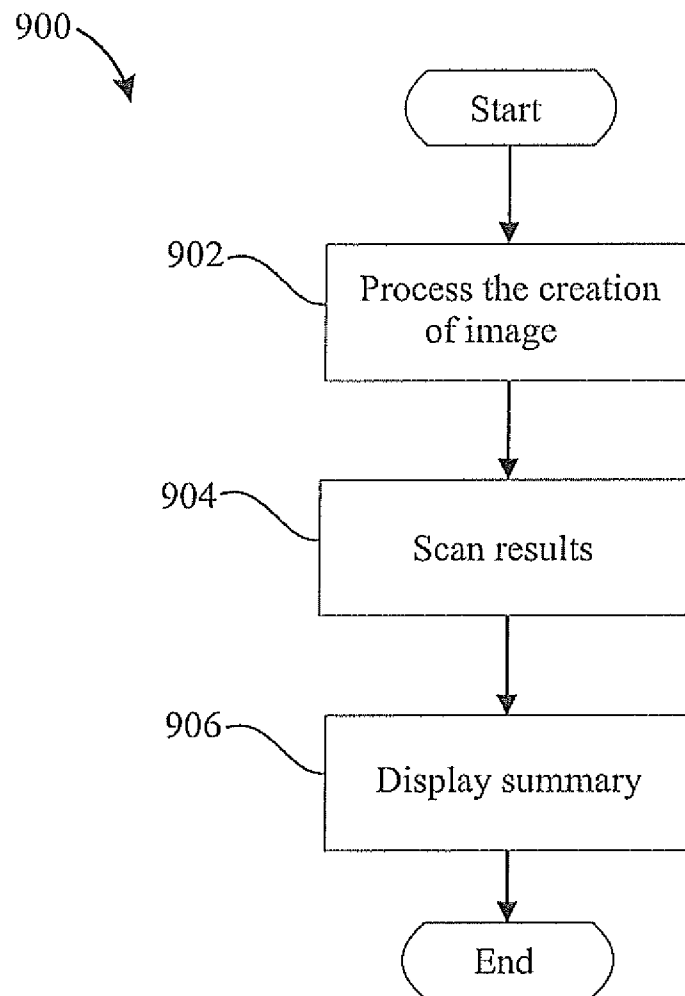


FIG. 9

A. CLASSIFICATION OF SUBJECT MATTER**G06F 8/61(2018.01)i, G06F 9/445(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 8/61; G06F 15/16; G06F 21/31; G06F 9/44; G06F 9/445; G06F 9/45; G06F 9/50; H04L 12/911

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: software container, image information, preference, negotiation, identify

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 9720709 B1 (AMAZON TECHNOLOGIES, INC.) 01 August 2017 See column 10, lines 8-26; column 16, line 53 - column 17, line 19; and figures 3, 8.	1-10
Y	US 2016-0026442 A1 (MAHESH KUMAR CHHAPARIA) 28 January 2016 See paragraphs [0024], [0027], [0059], [0075]; and figures 1-4.	1-10
A	WO 2016-077367 A1 (AMAZON TECHNOLOGIES, INC.) 19 May 2016 See paragraphs [0025]-[0037]; and figure 1.	1-10
A	KR 10-2015-0054496 A (SAMSUNG ELECTRONICS CO., LTD.) 20 May 2015 See claim 5; and figures 1-3.	1-10
A	KR 10-2016-0110460 A (ORACLE INTERNATIONAL CORPORATION) 21 September 2016 See claims 8-14; and figure 1.	1-10



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

20 December 2019 (20.12.2019)

Date of mailing of the international search report

20 December 2019 (20.12.2019)

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

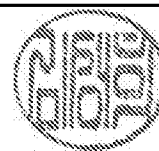


Facsimile No. +82-42-481-8578

Authorized officer

KIM, Sung Hee

Telephone No. +82-42-481-3516



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/MY2019/050042

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 9720709 B1	01/08/2017	US 9122562 B1	01/09/2015
US 2016-0026442 A1	28/01/2016	None	
WO 2016-077367 A1	19/05/2016	AU 2015-346530 A1	01/06/2017
		AU 2015-346530 B2	04/04/2019
		AU 2015-346530 C1	05/09/2019
		AU 2019-204805 A1	25/07/2019
		CN 107111519 A	29/08/2017
		EP 3218808 A1	20/09/2017
		JP 2017-538204 A	21/12/2017
		JP 2019-149192 A	05/09/2019
		JP 6522750 B2	29/05/2019
		KR 10-1952573 B1	27/02/2019
		KR 10-2017-0085072 A	21/07/2017
		KR 10-2019-0020843 A	04/03/2019
		KR 10-2042988 B1	11/11/2019
		RU 2018-130629 A	15/03/2019
		RU 2018-130629 A3	20/05/2019
		RU 2666475 C1	07/09/2018
		SG 11201703680W A	29/06/2017
		US 2016-0162320 A1	09/06/2016
		US 2019-108049 A1	11/04/2019
		US 9256467 B1	09/02/2016
		US 9996380 B2	12/06/2018
KR 10-2015-0054496 A	20/05/2015	US 2015-0135163 A1	14/05/2015
		US 9317282 B2	19/04/2016
KR 10-2016-0110460 A	21/09/2016	CN 105900059 A	24/08/2016
		CN 105900059 B	07/06/2019
		CN 106462466 A	22/02/2017
		CN 106462470 A	22/02/2017
		CN 106462717 A	22/02/2017
		CN 106462717 B	14/06/2019
		CN 106462833 A	22/02/2017
		CN 106471471 A	01/03/2017
		CN 106471472 A	01/03/2017
		CN 106575242 A	19/04/2017
		CN 106663035 A	10/05/2017
		CN 106716360 A	24/05/2017
		CN 107077382 A	18/08/2017
		CN 107077383 A	18/08/2017
		CN 107077388 A	18/08/2017
		CN 107077389 A	18/08/2017
		EP 3097481 A1	30/11/2016
		EP 3158441 A1	26/04/2017
		EP 3158442 A1	26/04/2017
		EP 3158443 A1	26/04/2017

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/MY2019/050042

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		EP 3158443 B1	17/04/2019
		EP 3158447 A1	26/04/2017
		EP 3158494 A1	26/04/2017
		EP 3161633 A1	03/05/2017
		EP 3167369 A1	17/05/2017
		EP 3170078 A1	24/05/2017
		EP 3198426 A1	02/08/2017
		EP 3198427 A1	02/08/2017
		EP 3198431 A1	02/08/2017
		EP 3198435 A1	02/08/2017
		EP 3198436 A1	02/08/2017
		JP 2017-503281 A	26/01/2017
		JP 2017-519306 A	13/07/2017
		JP 2017-519307 A	13/07/2017
		JP 2017-519308 A	13/07/2017
		JP 2017-519309 A	13/07/2017
		JP 2017-520058 A	20/07/2017
		JP 2017-520861 A	27/07/2017
		JP 2017-525035 A	31/08/2017
		JP 2017-526048 A	07/09/2017
		JP 2017-528845 A	28/09/2017
		JP 2017-528846 A	28/09/2017
		JP 2017-529628 A	05/10/2017
		JP 2017-532677 A	02/11/2017
		JP 2017-532678 A	02/11/2017
		JP 6461167 B2	30/01/2019
		JP 6491243 B2	27/03/2019
		JP 6505823 B2	24/04/2019
		JP 6510568 B2	08/05/2019
		JP 6542810 B2	10/07/2019
		KR 10-2017-0020866 A	24/02/2017
		KR 10-2017-0021874 A	28/02/2017
		KR 10-2017-0023112 A	02/03/2017
		KR 10-2017-0023121 A	02/03/2017
		KR 10-2017-0024014 A	06/03/2017
		KR 10-2017-0032360 A	22/03/2017
		KR 10-2017-0033350 A	24/03/2017
		KR 10-2017-0044639 A	25/04/2017
		KR 10-2017-0058955 A	29/05/2017
		KR 10-2017-0059449 A	30/05/2017
		KR 10-2017-0059451 A	30/05/2017
		KR 10-2017-0059454 A	30/05/2017
		KR 10-2017-0060036 A	31/05/2017
		US 10009225 B2	26/06/2018
		US 10015242 B2	03/07/2018
		US 10027550 B2	17/07/2018
		US 10027595 B2	17/07/2018
		US 10027716 B2	17/07/2018
		US 10038645 B2	31/07/2018
		US 10050903 B2	14/08/2018

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/MY2019/050042

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 10051043 B2	14/08/2018
		US 10084723 B2	25/09/2018
		US 10084843 B2	25/09/2018
		US 10091135 B2	02/10/2018
		US 10097589 B2	09/10/2018
		US 10103946 B2	16/10/2018
		US 10104011 B2	16/10/2018
		US 10178184 B2	08/01/2019
		US 10187454 B2	22/01/2019
		US 10193754 B2	29/01/2019
		US 10225209 B2	05/03/2019
		US 10250512 B2	02/04/2019
		US 10284486 B2	07/05/2019
		US 10318280 B2	11/06/2019
		US 10348565 B2	09/07/2019
		US 10348822 B2	09/07/2019
		US 10356161 B2	16/07/2019
		US 2015-0207758 A1	23/07/2015
		US 2015-370549 A1	24/12/2015
		US 2015-370608 A1	24/12/2015
		US 2015-372883 A1	24/12/2015
		US 2015-372887 A1	24/12/2015
		US 2015-372936 A1	24/12/2015
		US 2015-372937 A1	24/12/2015
		US 2015-372938 A1	24/12/2015
		US 2015-372941 A1	24/12/2015
		US 2015-373004 A1	24/12/2015
		US 2015-373097 A1	24/12/2015
		US 2015-373098 A1	24/12/2015
		US 2015-373099 A1	24/12/2015
		US 2015-373117 A1	24/12/2015
		US 2016-013983 A1	14/01/2016
		US 2016-014038 A1	14/01/2016
		US 2016-014191 A1	14/01/2016
		US 2016-014212 A1	14/01/2016
		US 2016-020949 A1	21/01/2016
		US 2016-085543 A1	24/03/2016
		US 2016-088108 A1	24/03/2016
		US 2016-092278 A1	31/03/2016
		US 2016-092319 A1	31/03/2016
		US 2016-092342 A1	31/03/2016
		US 2016-094385 A1	31/03/2016
		US 2016-094403 A1	31/03/2016
		US 2016-094404 A1	31/03/2016
		US 2016-094405 A1	31/03/2016
		US 2016-094406 A1	31/03/2016
		US 2016-094407 A1	31/03/2016
		US 2016-094408 A1	31/03/2016
		US 2016-094473 A1	31/03/2016
		US 2016-094474 A1	31/03/2016

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/MY2019/050042

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 2016-094478 A1	31/03/2016
		US 2016-094484 A1	31/03/2016
		US 2016-094486 A1	31/03/2016
		US 2016-094498 A1	31/03/2016
		US 2016-094510 A1	31/03/2016
		US 2016-094582 A1	31/03/2016
		US 2016-094583 A1	31/03/2016
		US 2016-094624 A1	31/03/2016
		US 2016-094625 A1	31/03/2016
		US 2016-094626 A1	31/03/2016
		US 2016-094627 A1	31/03/2016
		US 2016-094635 A1	31/03/2016
		US 2016-094647 A1	31/03/2016
		US 2016-142506 A1	19/05/2016
		US 2016-231998 A1	11/08/2016
		US 2016-328268 A1	10/11/2016
		US 2017-017494 A1	19/01/2017
		US 2017-019467 A1	19/01/2017
		US 2017-019485 A1	19/01/2017
		US 2017-034071 A1	02/02/2017
		US 2017-126742 A1	04/05/2017
		US 2017-192772 A1	06/07/2017
		US 2018-027023 A1	25/01/2018
		US 2018-165087 A1	14/06/2018
		US 2018-227242 A1	09/08/2018
		US 2019-014060 A1	10/01/2019
		US 2019-215279 A1	11/07/2019
		US 9405530 B2	02/08/2016
		US 9578009 B2	21/02/2017
		US 9697052 B2	04/07/2017
		US 9747137 B2	29/08/2017
		US 9792099 B2	17/10/2017
		US 9807119 B2	31/10/2017
		US 9843629 B2	12/12/2017
		US 9906601 B2	27/02/2018
		US 9916153 B2	13/03/2018
		US 9928111 B2	27/03/2018
		US 9930129 B2	27/03/2018
		US 9959421 B2	01/05/2018
		US 9961011 B2	01/05/2018
		US 9967324 B2	08/05/2018
		US 9971671 B2	15/05/2018
		US 9973384 B2	15/05/2018
		US 9977681 B2	22/05/2018
		WO 2015-112614 A1	30/07/2015
		WO 2015-200372 A1	30/12/2015
		WO 2015-200374 A1	30/12/2015
		WO 2015-200375 A1	30/12/2015
		WO 2015-200376 A1	30/12/2015
		WO 2015-200379 A1	30/12/2015

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/MY2019/050042Patent document
cited in search reportPublication
datePatent family
member(s)Publication
date

WO 2015-200474 A1	30/12/2015
WO 2016-007922 A1	14/01/2016
WO 2016-011051 A1	21/01/2016
WO 2016-049376 A1	31/03/2016
WO 2016-049576 A1	31/03/2016
WO 2016-049578 A1	31/03/2016
WO 2016-049582 A1	31/03/2016
WO 2016-049584 A1	31/03/2016