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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 April 2023 (13.04.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/059347 A1

(51) International Patent Classification:
G06F 9/455 (2006.01) *G06F 9/50* (2006.01)
G06F 9/48 (2006.01)

(21) International Application Number:
PCT/US2021/057184

(22) International Filing Date:
29 October 2021 (29.10.2021)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
17/493,010 04 October 2021 (04.10.2021) US

(71) Applicant: **UIPATH, INC.** [US/US]; 90 Park Avenue, 20th Floor, New York, New York 10016 (US).

(72) Inventors: **MA, Tao**; 90 Park Avenue, 20th Floor, New York, New York 10016 (US). **MAR, Joseph**; 90 Park Avenue, 20th Floor, New York, New York 10016 (US). **GUIDONE, Gregory Charles**; 90 Park Avenue, 20th Floor, New York, New York 10016 (US).

(74) Agent: **GRIFFIN, Partick S.**; Volpe Koenig, 30 S. 17th Street, Suite 1800, Philadelphia, Pennsylvania 19103 (US).

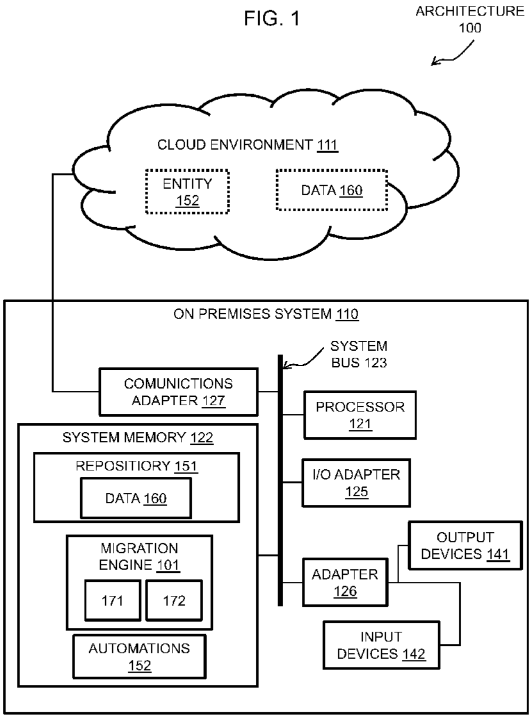
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, 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).

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

(54) Title: CLOUD MIGRATION

FIG. 1



(57) Abstract: Disclosed herein is a system. The system includes a memory and a processor. The memory stores processor executable instructions for a migration engine. The processor is coupled to the memory. The processor executes the migration engine to cause the system to implement an export operation for an on premises system to mine for data corresponding to automations or user specific arrangements. The processor, also, executes the migration engine to cause the system to implement an import operation of the data to a cloud environment to replicate the automations or user specific arrangements.

CLOUD MIGRATION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Application No. 17/493,010 filed October 4, 2021; the contents of which are incorporated herein by reference.

BACKGROUND

[0002] The disclosure herein relates to the field of cloud migration, and particularly to on premises data cloud migration.

[0003] Conventional operating and software systems require users to manually create settings, configurations, robots, triggers, machines, queues, webhooks, and other entities on premises (i.e., within the conventional operating and software systems). Then, when the users desire to migrate to a cloud environment, conventional cloud systems require users to manually recreate the same on-prem settings, configurations, robots, triggers, machines, queues, webhooks, and other entities in the cloud environment. Thus, there is a need for a cloud migration tool.

SUMMARY

[0004] According to one or more embodiments, a system is provided. The system includes a memory and a processor. The memory stores processor executable instructions for a migration engine. The processor is coupled to the memory. The processor executes the migration engine to cause the system to implement an export operation for an on premises system to mine for data corresponding to automations or user specific arrangements. The processor, also, executes the migration engine to cause the system to implement an import operation of the data to a cloud environment to replicate the automations or user specific arrangements.

[0005] According to one or more embodiments, the system embodiment above can be implemented as a method, an apparatus, a device, and/or a computer program product.

BRIEF DESCRIPTION OF THE DRAWING(S)

[0006] A more detailed understanding may be had from the following description, given by way of example in conjunction with the accompanying drawings, wherein like reference numerals in the figures indicate like elements, and wherein:

[0007] FIG. 1 depicts an architecture according to one or more embodiments;

[0008] FIG. 2 depicts examples of automations according to one or more embodiments;

[0009] FIG. 3 depicts a method according to one or more embodiments;

[0010] FIG. 4 depicts a method according to one or more embodiments;

[0011] FIG. 5 depicts an example user interface according to one or more embodiments;

[0012] FIG. 6 depicts an example user interface according to one or more embodiments;

[0013] FIG. 7 depicts an example user interface according to one or more embodiments;

[0014] FIG. 8 depicts a method according to one or more embodiments;

[0015] FIG. 9 depicts an example user interface according to one or more embodiments;

[0016] FIG. 10 depicts an example user interface according to one or more embodiments; and

[0017] FIG. 11 depicts an example user interface according to one or more embodiments.

DETAILED DESCRIPTION

[0018] The disclosure herein relates to the field of cloud migration, and particularly to on premises data cloud migration. For instance, on premises data cloud migration is implemented by a migration engine (i.e., a cloud migration tool) that includes software and/or hardware (e.g., processor executable code that is necessarily rooted in the hardware), such as robotic process automations (RPAs) or the like, that replicate existing on premises operating and software systems and corresponding data into a cloud environment. The migration engine, in turn, enables users to skip any manual recreation operations of automations via a user interface (UI) of the cloud environment. Accordingly, one or more technical effects, benefits, and advantages of the migration engine include enabling users to seamless migrate/move existing automations to the cloud environment without missing technical configurations and providing summaries of the migration/move that indicates successful and failed entity moves and data errors and warnings.

[0019] Turning now to FIG. 1, an architecture 100 is illustrated according to one or more embodiments. The architecture 100 can be representative of one or more computing devices, computing apparatuses, and/or computing environments, which comprise hardware, software, or a combination thereof supporting a migration engine 101 (i.e., a cloud migration tool). Further, embodiments of the architecture 100 and the migration engine 1010 disclosed herein may include one or more apparatuses, systems, methods, and/or computer program products at any possible technical detail level of integration.

[0020] As shown, the architecture 100 has an on premises system 110 and a cloud environment 111. The one premises system 110 can include one or more existing applications 119, one or more processors 121, one or more system memories 122, a system bus 123, one or more adapters 125, 126, and 127, one or more input devices 141, and one or more output devices 142. The system memory 122 can support at least a repository 151, the migration engine 101, and one or more automations 152. The repository 151 can store data 160. Further, based on operations 171 and 172 of the migration engine 101, the one or more automations 152 and the data 160 can be migrated to the cloud environment 111 (as represented by the dotted-outline). Note that while a single block is shown for any one element of the architecture 100, the single block is representative of one or more of that element. Note, also, that any element of the architecture 100 may be referred to in the singular or in the plural herein. Further, any of the elements of the on premises system 110 can also be found in the cloud environment 111 as needed for proper operation of the migration engine 101.

[0021] The on premises system 110 can be any computing device or system adapted or configured to perform as an online platform, a server, an embedded computing system, a personal computer, a console, a personal digital assistant (PDA), a cell phone, a tablet computing device, a quantum computing device, cloud computing device, a mobile device, a smartphone, a fixed mobile device, a smart display, a wearable computer, or the like. On premises, which is also referred to as "on-premise," "on-premises," or "on-prem," is a location of deployed software, such as within the computing device or the system. In contrast to on premises, off premises can anywhere external to the computing device or the system, such as in the cloud environment 111. The on premises system 110 can, further, provide the one or more existing applications 119 that utilizes the automations 152 and the data 160. The one or more existing applications 119 is software that performs specific tasks, such as a orchestration software (i.e., On-Prem Orchestrator) that provides automated configuration, coordination, and management of the on premises system 110, as well as word processor, graphic, database, spreadsheet, presentations, web browser, and enterprise software.

[0022] The cloud environment 111 can be any computing environment external to the on premises system 110. The cloud environment 111 can provide on-demand availability of resources therein, especially data storage and computing power, distributed over multiple physical locations (e.g., one or more data centers). The cloud environment 111 can, further, provide a replication of any application that are within the on premises system 110, such as a cloud organization or a Cloud Orchestrator that mirrors the existing application 119 or the On-Prem Orchestrator.

[0023] The processor 121 may be any type of general or specific purpose processor, including a central processing unit (CPU), application specific integrated circuit (ASIC), field programmable gate array (FPGA), graphics processing unit (GPU), controller, multi-core processing unit, three dimensional processor, quantum computing device, or any combination thereof. The processor 121 may also have multiple processing cores, and at least some of the cores may be configured to perform specific functions. Multi-parallel processing may also be configured with the processor 121. The processor 121, also referred to as a processing circuit, is coupled via the system bus 123 to a system memory 122 and various other components.

[0024] The system memory 122 is an example of a (non-transitory) computer readable storage medium, where the repository 151, the migration engine 101, and the automation 152 can be located/managed/stored. Further, the system memory 122 can include any combination of a read only memory (ROM), a random access memory (RAM), internal or external Flash memory, embedded static-RAM (SRAM), solid-state memory, cache, static storage, or any other types of volatile or non-volatile memory. Non-transitory computer readable storage mediums may be any media that can be accessed by the processor 121 and may include any media that is removable, non-removable, or the like.

[0025] The system bus 123 is a wiring or other mechanism configured for communicating information (e.g., the data 160) between the processor 121, the system memory 122, and various other components.

[0026] With respect to the adapters 125, 126, and 127, the on premises system 110 can particularly include an input/output (I/O) adapter 125, a device adapter 126, and a communications adapter 127.

[0027] According to one or more embodiments, the I/O adapter 125 can be configured as a small computer system interface (SCSI), of in view of frequency division multiple access (FDMA) single carrier FDMA (SC-FDMA), time division multiple access (TDMA), code division multiple access (CDMA), orthogonal frequency-division multiplexing (OFDM), orthogonal frequency-division multiple access (OFDMA), global system for mobile (GSM) communications, general packet radio service (GPRS), universal mobile telecommunications system (UMTS), cdma2000, wideband CDMA (W-CDMA), high-speed downlink packet access (HSDPA), high-speed uplink packet access (HSUPA), high-speed packet access (HSPA), long term evolution (LTE), LTE Advanced (LTE-A), 802.11x, Wi-Fi, Zigbee, Ultra-WideBand (UWB), 802.16x, 802.15, home Node-B (HnB), Bluetooth, radio frequency identification (RFID), infrared data association (IrDA), near-field communications (NFC), fifth generation (5G), new radio (NR), or any other wireless or wired device/transceiver for communication.

[0028] The device adapter 126 interconnects input/output devices 141 and 142 to the system bus 123, such as a display, a keyboard, a control device, or the like (e.g., a camera, a speaker, etc.).

[0029] The communications adapter 126 interconnects the system bus 123 with the could environment 111, which may be an through a network, enabling the on premises system 110 to communicate the automations 152 and the data 160.

[0030] The adapters 125, 126, and 127 may be connected to one or more I/O buses that are connected to the system bus 123 via an intermediate bus bridge. Suitable I/O buses for connecting peripheral devices, such as hard disk controllers, network adapters, and graphics adapters typically include common protocols, such as the Peripheral Component Interconnect (PCI).

[0031] The input device 141 can be an control device, such as a keyboard, a computer mouse, a touchpad, a touch screen, a keypad, or the like, for input to the on premises system 110. In addition, one or more inputs may be provided to the on premises system 110 remotely via another computing system in communication therewith, or the on premises system 110 may operate autonomously.

[0032] The output device 142 is configured to provide one or more UIs or graphic UIs (GUIs), which can be generated, captured, and analyzed by the migration engine 101. Examples of the output device 142 can include, but are not limited to, a plasma, a liquid crystal display (LCD), a light emitting diode (LED), a field emission display (FED), an organic light emitting diode (OLED) display, a flexible OLED display, a flexible substrate display, a projection display, a 4K display, a high definition (HD) display, a Retina© display, an in-plane switching (IPS) display or the like. The output device 142 may be configured as a touch, three dimensional (3D) touch, multi-input touch, or multi-touch display using resistive, capacitive, surface-acoustic wave (SAW) capacitive, infrared, optical imaging, dispersive signal technology, acoustic pulse recognition, frustrated total internal reflection, or the like as understood by one of ordinary skill in the art for input/output (I/O).

[0033] The repository 151 can be any database (e.g., an SQL database) and/or another storage mechanism. According to one or more embodiments, the repository 151 can include the data 160 as described herein for access by the migration engine 101. The data 160 can include raw unstructured and/or structured data created or used by the automations 152.

Example of data 160 include, but are not limited to, images, segments, hashes, video, frames, source data, robot video, historical data, and source code. According to one or more embodiments, the data 160 can be user specific arrangements including settings, configurations, triggers, and preferences.

[0034] The automations 152 can be representative robots, machines, webhooks, scripts, applications macros, on-prem settings, configurations, triggers, queues, and other entities. Turning to FIG. 2, examples of the automations 151 are depicted according to one or more embodiments.

[0035] FIG. 2 also shows a system memory 200 storing one or more robots 201, 202, and 203. The system memory 200 is an example of the system memory 122 of FIG. 1. Generally, the migration engine 101 can include or operate in conjunction with a studio component that produces the one or more robots 201, 202, and 203 that perform one or more activities 222 and/or provide one or more UI automations 224 using one or more driver components 226 and/or internal engines 228. The one or more robots 201, 202, and 203 may also engage with other driver components 230 and modules 240.

[0036] For example, the robot 201 (as well as the attended robot 202 and the unattended robots 203) can be an application, an applet, or the like that may perform and/or automate one or more workflows (e.g., represented in part by the one or more activities 222 and/or provide the one or more UI automations 224). A workflow may include, but are not limited to, task sequences, flowcharts, Finite State Machines (FSMs), global exception handlers, UI transparents, or the like. Task sequences may be linear processes for handling linear tasks between one or more applications or windows. Flowcharts may be configured to handle complex business logic, enabling integration of decisions and connection of activities in a more diverse manner through multiple branching logic operators. FSMs may be configured for large workflows. FSMs may use a finite number of states in their execution, which may be triggered by a condition, transition, activity, or the like. Global exception handlers may be configured to determine workflow behavior when encountering an execution error, for debugging processes, or the like. UI transparents can be software operations to an underlying operating system (OS) or hardware. Non-limiting examples of operations that can be accomplished by one or more workflows may be one or more of performing log-ins, filling a form, information technology (IT) management, or the like.

[0037] Further, for attended operations, automation by the attended robot 202 may be performed by receiving input, commands, instructions, guidance, or the like from a third party component. For unattended operations, automation by the unattended robot 203 may be performed without third party inputs or control.

[0038] According to one or more embodiments, the one or more robots and the executor 170 201, 202, and 203 may be split, distributed, or the like into several components, each being dedicated to a particular automation task or activity. Robot components may include SCM-managed robot services, user mode robot services, executors, agents, command line, or the like. SCM-managed robot services may manage or monitor Windows® sessions and act as a proxy for an execution hosts (i.e., the on premises system 110 on which the one or more robots 201, 202, and 203 are executed). These services may be trusted and managed with credentials.

[0039] According to one or more embodiments, the activities 222 and/or the UI automations 224 may be performed by the robot 201 using the drivers 226 therein, the driver components 230, and/or the modules 240. The drivers 226 may be utilized for UI automation by the robot 201 to get elements of a UI. The driver components 230 may include, but are not limited to, OS drivers, browser drivers, virtual machine drivers, enterprise drivers, and the like. In certain configurations, the modules 240 may be a driver used for UI automation. Further, any of these actions by the robot 201 can be implemented by the attended robots 202 and the unattended robots 203.

[0040] Further, FIG. 2 shows a system memory 200 storing machines 250, webhooks 260, scripts 270, applications macros 280, and other entities 290.

[0041] The machine 250 can include any mathematical model that defines an algorithm and or logic.

[0042] The webhook 260 can include any mechanism or interface for augmenting or altering a behavior of a web page or application with custom callbacks. Callbacks can be maintained, modified, and managed by users/developers for operation within the on premises system 110 of FIG. 1.

[0043] The script 270 can include any list of commands executed by software to automate processes on the on premises system 110 of FIG. 1.

[0044] The applications macro 280 can include any built-in application feature for carrying out a sequence and enabling recordation of the sequence and/or result.

[0045] The entities 290 can include identifiable and separate objects, bits of the data 160 of FIG. 1, user specific arrangements, and/or distinct system components. Other examples of entities 290 include, but are not limited to, packages, libraries, folders, calendars, environments, triggers, assets, metadata, etc.

[0046] Each of the one or more robots 201, 202, and 203, the machines 250, the webhooks 260, the scripts 270, the applications macros 280, and the entities 290 can operate based on user configurations, manually created settings, configurations, triggers, etc.

[0047] For example, to run a workflow for UI automation, the robot 201 may need to uniquely identify specific screen elements, such as buttons, checkboxes, text fields, labels, etc., regardless of application access or application development. Examples of application access may be local, virtual, remote, cloud, Citrix®, VMWare®, VNC®, Windows® remote desktop, virtual desktop infrastructure (VDI), or the like. Examples of application development may be win32, Java, Flash, hypertext markup language (HTML), HTML5, extensible markup language (XML), JavaScript, C#, C++, Silverlight, or the like.

[0048] According to one or more embodiments, at development, the robot 201 can be produced at/by a designer (e.g., in a studio software). Further, at deployment, the robot 201 may be managed, controlled, configured or the like at/by a conductor (e.g., within the on premises system 110 of FIG. 1). At deployment, the robot 201 may also be executed within an executor (e.g., within the on premises system 110 of FIG. 1). In an example, a user input (e.g., a selection on a display) can be submitted by a user or an operator to the designer 110. From the selection, which the designer 110 may provide to the robot 122, the robot 122 may determine representative data of area(s) of a visual display (e.g., selected by the user or the operator).

[0049] The migration engine 101 can be stored as software components, modules, engines, instructions, or the like for execution by the processor 121 to cause the on premises system 110 to operate, such as described herein with reference to FIGS. 3-10. For instance, the migration engine 101 can provide a specific configuring and tooling robotic process automation method with respect to an intelligent migration experience. The intelligent migration experience can include using robot intelligence (e.g., machine learning and artificial intelligence) to perform background monitoring/analyzing of user activity for purposes of identifying, collecting/harvesting, and moving the automations 152 and the data 160. Thus, the migration engine 101 can be a self-serve desktop application, requiring no code to provide a low barrier of entry that enables an average business user to use it.

[0050] In accordance with one or more embodiments, the migration engine 101 can be part of a framework/mechanism (e.g., the architecture 100) that automatically implements codes and designs of the operations 171 and 172 to provide application integration, along with automating of third-party applications, administrative information technology (IT) tasks, and/or business IT processes. Further, the migration engine 101 can include robot intelligence and/or can instruct the automations 152 to execute or monitor a workflow in a mainframe, web, virtual machine, remote machine, virtual desktop, enterprise platform, online platform, desktop app(s), browser, or the like client, application, or program. Furthermore, the migration engine 101 can be configured for provisioning, deployment, configuration, queueing, monitoring, logging, and/or providing interconnectivity. Provisioning may include creating and maintenance of connections or communication between the one or more robots, the on premises system 110, and the cloud environment 111. Deployment may include assuring the delivery of package versions to assigned robots for execution. Configuration may include maintenance and delivery of robot environments and process configurations. Queueing may include providing management of queues and queue items. Monitoring may include keeping track of robot identification data and maintaining user permissions. Logging may include storing and indexing logs to the repository 151 (e.g., an SQL database) and/or another storage mechanism (e.g., ElasticSearch®, which provides the ability to store and quickly query large datasets). The migration engine 101 may, also, provide interconnectivity by acting as the centralized point of communication for third-party solutions and/or applications.

[0051] According to one or more embodiments, the migration engine 101 can be configured in hardware, software, or a hybrid implementation. The migration engine 101 can be composed of modules that are in operative communication with one another, and to pass information or instructions.

[0052] According to one or more embodiments, the migration engine 101 can provide one or more user interfaces, such as on behalf of the on premises system 110, the cloud environment 111, and/or other application, as well as directly as needed to provide the intelligent migration experience. The user interfaces include, but are not limited to, internet browsers, graphic user interfaces, window interfaces, and/or other visual interfaces for applications, operating systems, file folders, and the like. Thus, user activity can include any interaction or manipulation of the user interfaces provided by the migration engine 101. The migration engine 101 can further include custom modules to perform application specific processes or derivatives thereof, such that the on premises system 110 may include additional functionality. For example, according to one or more embodiments, the migration engine 101 may be configured to store information, instructions, commands, or data to be executed or processed by the processor 121 to enable the operations 171 and 172 (i.e., an import and export operations as described with respect to FIG. 3).

[0053] Turning now to FIG. 3, a method 300 is depicted according to one or more embodiments. The method 300 is described with respect to FIGS 1-2, as implemented by the migration engine 101 of FIG. 1 (which is being executed by the one or more processors 121 within the on premises system 110). Generally, the method 300 overcomes the manually recreation process by automatically and intelligently migrating from the on premises system 119 to the cloud environment 110. For instance, the method 300 of the migration engine 101 takes an existing application 119 (e.g., an On-Prem Orchestrator) of the on premises system 110 and replicates the data 160 thereof into the could environment 111 (e.g., a cloud organization or a Cloud Orchestrator). This enables users of the existing application 119 to move their existing automations 152 and setup to the cloud environment 111 seamlessly, while receiving a brief summary of the migration indicating successes and failures.

[0054] The method 300 begins at block 310, where the migration engine 101 implements an export operation (i.e., the operation 171 of FIG. 1). More particularly, the migration engine 101 implements the export operation for the on premises system 110. The export operation mines for the data 160 corresponding to the existing application 119, the automations 152, and/or user specific arrangements (e.g., extract copies of the entities 290). According to one or more embodiment, the migration tool 101 includes a full application programable interface (API) to integrate with the existing application 119, detect dependencies between the entities 290 (i.e., parses through a validation layer of the entities), and detect errors with respect to those dependencies. In contrast to conventional operating and software systems and manual migration, generic scripts (i.e., generic programs or sequences of instructions) cannot detect such dependencies or errors.

[0055] According to one or more embodiments, the import operation can provide a first prompt or window (e.g., a user interface) requiring one or more credentials, such as a tenant name or a username and requiring a password. The migration engine 101, in conjunction with the export operation, can generate an entry list (e.g., provided via a user interface) that presents a summary of the data 160 mined.

[0056] The method 300 continues at block 330, where the migration engine 101 implements an import operation (i.e., the operation 172 of FIG. 1). The import operation migrates/moves the data 160 to the cloud environment 111 to replicate the automations 152 or user specific arrangements (e.g., recreates the entities 290). As shown in FIG. 1, based on operations 171 and 172 of the migration engine 101, the migrated automations 152 and the data 160 are represented by the dotted-outline.

[0057] The migration engine 101, in conjunction with the import operation, can generate an entry list (e.g., provided via a user interface) that presents a summary of the data 160 replicated, as well as an error list corresponding to call rejections and warning of dependencies issues. According to one or more embodiments, the import operation by the migration engine 101 can include a piece meal migration. The piece meal migration can determine what portions of the data 160 exists on the cloud environment 111 and can add only non-existing portions of the data 160.

[0058] According to one or more embodiments, the migration engine 101 and the method 300 can be implemented in both air gapped environment and connected environments, with respect to the on premises system 100. For example, air gapped environment can be privately run, not exposed to internet, which requires a physically move to between computers/systems. In turn, the migration engine 101 can export the data 160 while the on premises system 100 is configured as an air gapped environment, can store the exported data 160 locally (e.g., in the repository 151), and can import the stored data 160 to the cloud environment 111 once connected.

[0059] Turning now to FIG. 4, a method 400 is depicted according to one or more embodiments. The method 400 is described with respect to an export operation (i.e., the operation 171 of FIG. 1), as implemented by the migration engine 101 of FIG. 1.

[0060] The method 400 begins at block 410, where the migration engine 101 presents a (first) prompt. In this regard, the output devices 141 can provide or display the first prompt for a user to view. The first prompt can request one or more credentials, and can include at least a mechanism for confirming the initiation of the export operation, such as a “start” button.

[0061] FIG. 5 depicts an example user interface 500 (e.g., the first prompt) according to one or more embodiments. The example user interface 500, as generated by the migration engine 101, requests the one or more credentials, such as a uniform resource locator (URL) 510, a tenant name 520, a username 530, and/or a password 540, as well as provides a “start” button 550 and a “cancel” button 560.

[0062] The URL 510 is a location or address identifying where the existing application 119 (e.g., the On-Prem Orchestrator) of the on premises system 110 and the corresponding data 160 can be found. The tenant name 520 is an identification for a group of users who share a common access with specific privileges to the existing application 119, the automations 151, and the data 160. The username 530 is an identification of a user of the group who utilizes the automations 151 and the data 160. The password 540 is a string of alpha-numeric characters for authenticating the user. The “start” button 550 can initiate the export operation. The “cancel” button 560 exit the example user interface 500.

[0063] At block 420, the migration engine 101 receives the one or more credentials through the first prompt. In this regard, the user can engage the input device 142 to input the credentials requested by the example user interface 500.

[0064] At block 430, the migration engine 101 receives a command (e.g., click of the “start” button 550) to initiate the export operation (e.g., start exporting data from On-Prem Orchestrator).

[0065] At block 440, the migration engine 101 presents a progress of the export operation in a second prompt. In this regard, the output devices 141 can provide or display the second prompt for the user to view. The second prompt can present a loading screen indicating a progress of the export operation. Turning to FIG. 6, an example user interface 600 is depicted as the second prompt according to one or more embodiments. The example user interface 600, as generated by the migration engine 101, includes a loading screen indicating an exportation of the data 160 from the on premises system 110. The example user interface 600 can include a progress bar 685 and can indicate which entity is currently being exported at a point in time.

[0066] At block 450, the migration engine 101 present a third prompt. In this regard, the output devices 141 can provide or display the third prompt for the user to view. The third prompt can present summary of successfully imported entities (e.g., an exported entity list) to the user to view. According to one or more embodiments, the exported entity list itemizes the data 160 corresponding to the automations 152 or the user specific arrangements from the on premises system 110. The exported data 160 can be stored within the repository 151. Turning to FIG. 7, an example user interface 700 is depicted as the third prompt according to one or more embodiments. The example user interface 700 include a window 705, a “home” button 770, and a “open file” button 780. The window 705 can provide the exported entity list. As shown, the exported entity list can include a number of total entities exported, packages, libraries, calendars, folders, machines, environments, robots, triggers, assets, processes, queues, etc. The migration engine 101 can save the exported entity list as a summary file in the repository 151, such as in a text file format. That is, the exported entity list corresponds to all exported data from the existing application 119 and can be stored locally in the repository 151, such as in serialized JavaScript Object Notation (JSON) format, which is open standard file format and data interchange format. The “home” button 770 can return to the migration engine 101 to the first prompt or advance the migration engine 101 to the import operation as described with respect to FIG. 8. The “open file” button 780 can enable further viewing of any selected item of the exported entity list.

[0067] Turning now to FIG. 8, a method 800 is depicted according to one or more embodiments. The method 800 is described with respect to an import operation (i.e., the operation 172 of FIG. 1), as implemented by the migration engine 101 of FIG. 1. The method 800 begins at block 850, where the migration engine 101 presents a prompt or first window (e.g., a user interface). In this regard, the output devices 141 can provide or display the first window for a user to view. The first window can request one or more credentials, and can include at least a mechanism for confirming the initiation of the import operation, such as a "start" button. In this regard, the migration engine 101 can open a newly created tenant, where the user desires to migrate the exported data 160. The migration engine 101 can receive an API access key or user key, an account name, a tenant name, and a client identification (ID) and copy these details for importing the exported data 160 to the newly created tenant.

[0068] FIG. 9 depicts an example user interface 900 (e.g., the first window) according to one or more embodiments. The example user interface 900, as generated by the migration engine 101, requests the one or more credentials, such as an account name 925, a tenant name 935, and/or a client ID 945, as well as provides a "start" button 950 and a "cancel" button 960.

[0069] The account name 925 is a unique URL site. The tenant name 935 is an identification for a group of users who share a common access with specific privileges to the existing application 119, the automations 151, and the data 160. The client ID 945 is an identification of a user of the group who utilizes the automations 151 and the data 160. The client ID 945 can be specific to the existing application 119 itself and be the same for all users and tenants on the on premises system 110.

[0070] The “start” button 950 can initiate the import operation. That is, the “start” button 950 can be a “Start Import” button that, when pressed, causes the migration engine 101 to present a login screen (e.g., that points to the account name 925). Once logged-in (e.g., using an account associated), the migration engine 101 verifies that the import operation is authorized to run and the import operation resumes. The “cancel” button 960 exit the example user interface 900.

[0071] Optionally, according to one or more embodiments, a user key field can be included in the example user interface 900. The user key field can be an encrypted string that identifies an application (e.g., the cloud organization or the Cloud Orchestrator that mirrors the existing application 119 or the On-Prem Orchestrator). The migration engine 101 can generate unique login keys (i.e., the user key enables) used with APIs to log in and perform actions on a user’s behalf.

[0072] At block 870, the migration engine 101 receives the one or more credentials through the first window. In this regard, the user can engage the input device 142 to input the credentials requested by the example user interface 900.

[0073] At block 880, the migration engine 101 receives a command (e.g., click of the “start” button 950) to initiate the import operation (e.g., start import data to the Cloud Orchestrator).

[0074] At block 890, the migration engine 101 presents a progress of the import operation in a second window. In this regard, the output devices 141 can provide or display the second window for the user to view. The second window can present a downloading screen indicating a progress of the import operation.

[0075] Turning to FIG. 10, an example user interface 1000 is depicted as the second window according to one or more embodiments. The example user interface 1000, as generated by the migration engine 101, includes a downloading screen indicating an importation of the data 160 to the cloud environment 111. The example user interface 1000 can include a progress bar 1085 and can indicate which entity is currently being imported at a point in time.

[0076] At block 895, the migration engine 101 present a third window. In this regard, the output devices 141 can provide or display the third window for the user to view. The third window can present summary of successfully imported and failed entities (e.g., an importation list). According to one or more embodiments, the importation list itemizes the data 160 corresponding to the automations 152 or the user specific arrangements from the on premises system 110 that was imported and/or failed. Turning to FIG. 11, an example user interface 1100 is depicted as the third window according to one or more embodiments. The example user interface 1100 include a window 1105, a “view” button 1170, and a “done” button 1180. The window 1105 can provide the importation list. As shown, the importation list can include a number of total entities attempted, imported, and note imported, as well as a number for each entity. The “view” button 1170 can enable further viewing of any selected item of the importation list. The “done” button 1180 can exit the example user interface 1100.

[0077] Further, this data 160 and the importation list can be stored within the repository 151, such as in text file format as described herein. Each failed entity of the importation list can correspond to an errors related to API call rejection while importing. Each failed entity of the importation list can further include one or and the warnings are related to dependencies, which in turn can be rectified by a manual check. For instance, errors can occur on an import operation due to business validations. In some cases, the migration engine 101 can make calls and determinations so that the errors can be resolved automatically. For example, if a schedule is dependent on a package and the package is not exported and/or uploaded, the migration engine 101 can detect the dependency of schedule on package and write a warning or directly identify, copy, and import the package to resolve the dependency. Additionally, while importing, entities that are not sequenced randomly because of their dependencies may require a document for indicating specific order of entity folders for cloud migration. The migration engine 101 can generate this document and outline rules for all the dependencies. For example, if there is a job that uses a package, then the package has to be first and the migration engine 101 can create a rule that imports that package first (according to the document).

[0078] According to one or more embodiments, the migration engine 101 can implement a piece meal migration for the cloud tenant when some of the data already exists. In this regard, when the import operation is performed, the migration engine 101 determines what data exists already the cloud environment 111 and only adds data that is not yet existing.

[0079] The flowchart and block diagrams in the drawings illustrate the architecture, functionality, and operation of possible implementations of apparatuses, systems, methods, and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). In some alternative implementations, the functions noted in the blocks may occur out of the order noted in the flowchart and block diagrams in the drawings. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

[0080] Although features and elements are described above in particular combinations, one of ordinary skill in the art will appreciate that each feature or element can be used alone or in any combination with the other features and elements. For instance, for any of the methods and processes described herein, the steps recited may be performed out of sequence in any order and sub-steps not explicitly described or shown may be performed. When using referring to “A or B”, it may include A, B, or A and B, which may be extended similarly to longer lists. When using the notation X/Y it may include X or Y. Alternatively, when using the notation X/Y it may include X and Y. X/Y notation may be extended similarly to longer lists with the same explained logic. In addition, "coupled" or "operatively coupled" may mean that objects are linked but may have zero or more intermediate objects between the linked objects. Also, any combination of the disclosed features/elements may be used in one or more embodiments.

[0081] In addition, the methods and processes described herein may be implemented in a computer program, software, and/or firmware (e.g., a computer program product) incorporated in a computer-readable medium for execution by a computer or processor. That is, the computer program product may include a computer readable storage medium (or media) having computer readable program instructions thereon for causing a controller, processor, or the like to carry out aspects of the present invention.

[0082] The computer readable storage medium can be a tangible device that can retain and store computer readable program instructions. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. The computer readable storage medium, as used herein, is not to be construed as being transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire. Examples of computer-readable storage media include, but are not limited to, a register, cache memory, semiconductor memory devices, magnetic media such as internal hard disks and removable disks, magneto-optical media, optical media such as compact disks (CD) and digital versatile disks (DVDs), a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a static random access memory (SRAM), and a memory stick.

[0083] The computer readable program instructions described herein can be communicated and/or downloaded to respective controllers, processors, or the like from an apparatus, device, computer, or external storage via a connection, for example, network communications. Computer readable program instructions for carrying out operations of the present invention may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, configuration data for integrated circuitry, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language such as Smalltalk, C++, or the like, and procedural programming languages, such as the "C" programming language or similar programming languages. In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present invention.

[0084] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used herein, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one more other features, integers, steps, operations, element components, and/or groups thereof.

[0085] The descriptions of the various embodiments herein have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments disclosed herein.

CLAIMS

What is claimed:

1. A system comprising:

a memory configured to store processor executable instructions for a migration engine; and

at least one processor coupled to the memory and configured to execute the processor executable to cause the system to:

implementing, by the migration engine, an export operation for an on premises system to mine for data corresponding to automations or user specific arrangements;

implementing, by the migration engine, an import operation of the data to a cloud environment to replicate the automations or user specific arrangements.

2. The system of claim 1, wherein the user specific arrangements comprises settings, configurations, triggers, and preferences.

3. The system of claim 1, wherein the automations comprises robots, machines, webhooks, scripts, applications macros, and entities.

4. The system of claim 1, wherein the export operation provides a prompt requiring a tenant name or a username and requiring a password.

5. The system of claim 1, wherein the migration engine generates an entry list presenting a summary of the data mined and replicated.

6. The system of claim 1, wherein the migration engine implements a piece meal migration, during the import operation, that determines what portions of the data exists on the cloud environment and adds only non-existing portions of the data.

7. The system of claim 1, wherein the system comprises an air gapped environment and a connected environment.

8. The system of claim 1, wherein the migration engine generates an error list corresponding to call rejections and warning of dependencies issues.

9. The method of claim 8, wherein items of an error list are configured to be viewed and rectified to complete the import of the items.

10. The system of claim 1, wherein the data corresponding to the automations or the user specific arrangements from the on premises system is stored with a repository of the on premises system in serialized JSON format.

11. A method implemented by a migration engine stored as processor executable instructions in a memory coupled to at least one processor, the method comprising:

implementing an export operation for an on premises system to mine for data corresponding to automations or user specific arrangements; and

implementing an import operation of the data to a cloud environment to replicate the automations or the user specific arrangements.

12. The method of claim 11, wherein the user specific arrangements comprises settings, configurations, triggers, and preferences.

13. The method of claim 11, wherein the automations comprises robots, machines, webhooks, scripts, applications macros, and entities.

14. The method of claim 11, wherein the export operation provides a prompt requiring a tenant name or a username and requiring a password.

15. The method of claim 11, wherein the migration engine generates an entry list presenting a summary of the data mined and replicated.

16. The method of claim 11, wherein the migration engine implements a piece meal migration, during the import operation, that determines what portions of the data exists on the cloud environment and adds only non-existing portions of the data.

17. The method of claim 11, wherein the system comprises an air gapped environment and a connected environment.

18. The method of claim 11, wherein the migration engine generates an error list corresponding to call rejections and warning of dependencies issues.

19. The method of claim 18, wherein items of an error list are configured to be viewed and rectified to complete the import of the items.

20. The method of claim 11, wherein the data corresponding to the automations or the user specific arrangements from the on premises system is stored with a repository of the on premises system in serialized JSON format.

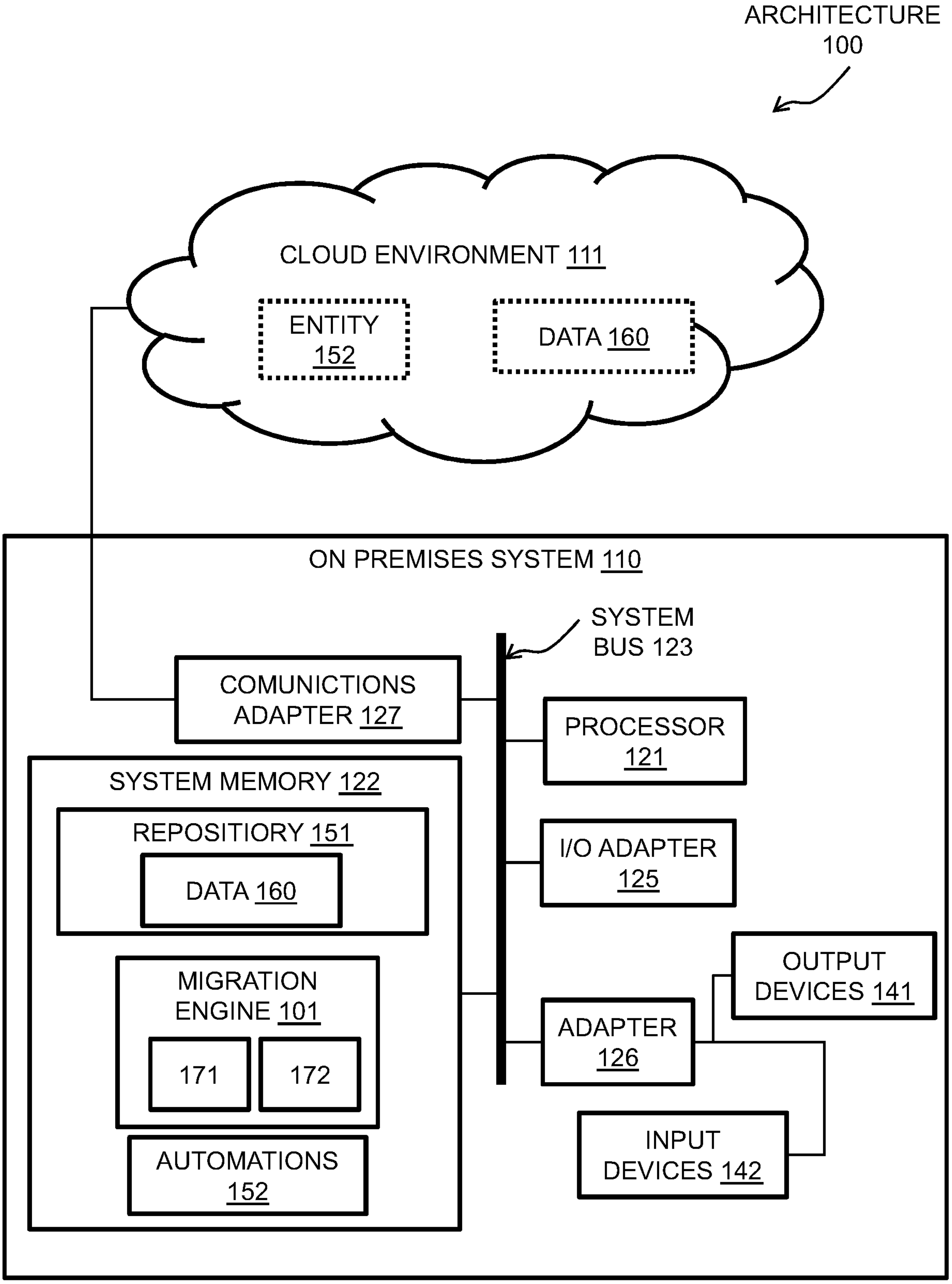

ARCHITECTURE 100
CLOUD ENVIRONMENT 111
ENTITY 152
DATA 160
ON PREMISES SYSTEM 110
SYSTEM BUS 123
COMUNICTIONS ADAPTER 127
PROCESSOR 121
SYSTEM MEMORY 122
REPOSITIORY 151
DATA 160
I/O ADAPTER 125
OUTPUT DEVICES 141
MIGRATION ENGINE 101
171
172
ADAPTER 126
INPUT DEVICES 142
AUTOMATIONS 152

FIG. 1

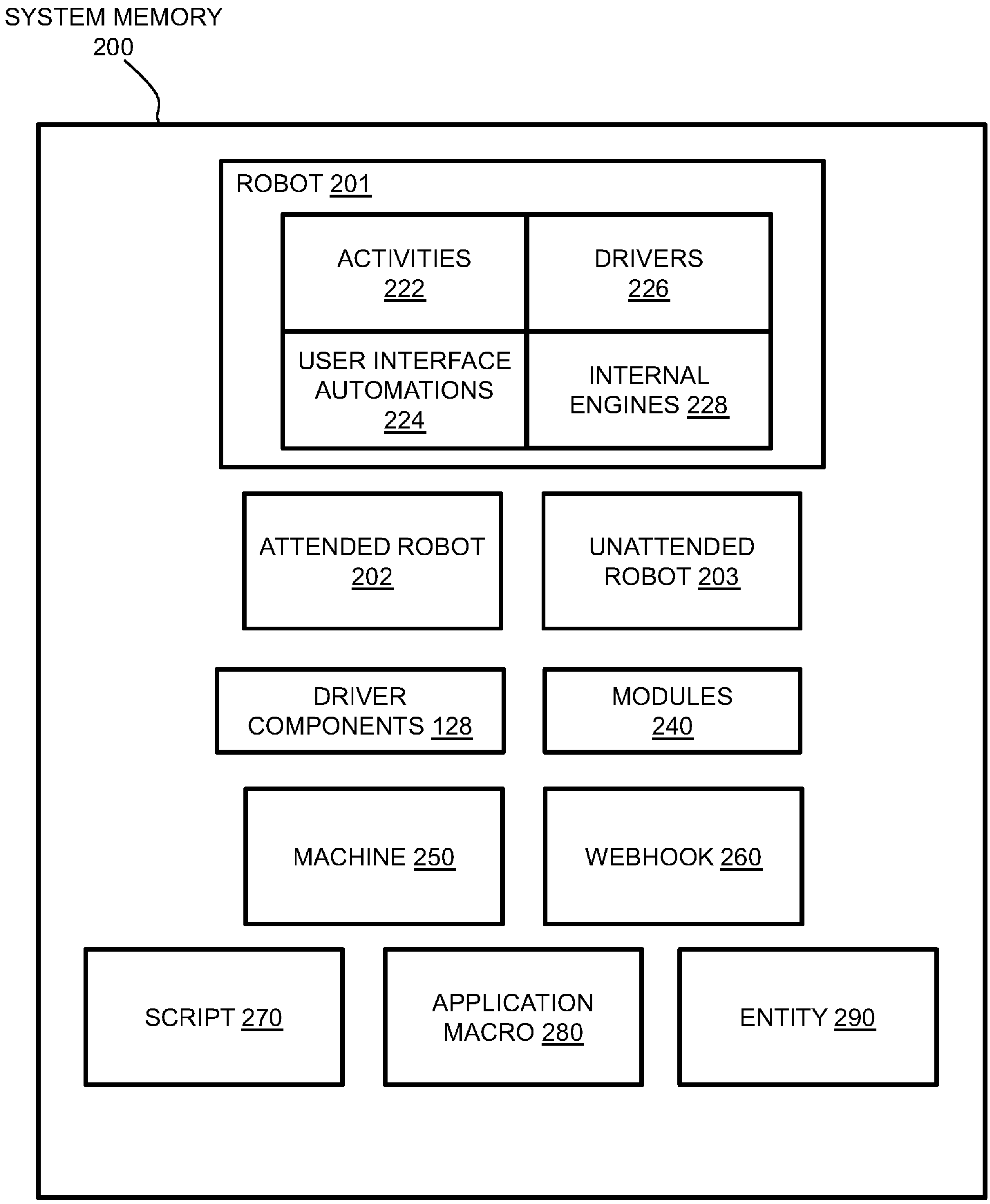

SYSTEM MEMORY
200
ROBOT 201
ACTIVITIES
222
DRIVERS
226
USER INTERFACE
AUTOMATIONS
224
INTERNAL
ENGINES 228
ATTENDED ROBOT
202
UNATTENDED
ROBOT 203
DRIVER
COMPONENTS 128
MODULES
240
MACHINE 250
WEBHOOK 260
SCRIPT 270
APPLICATION
MACRO 280
ENTITY 290

FIG. 2

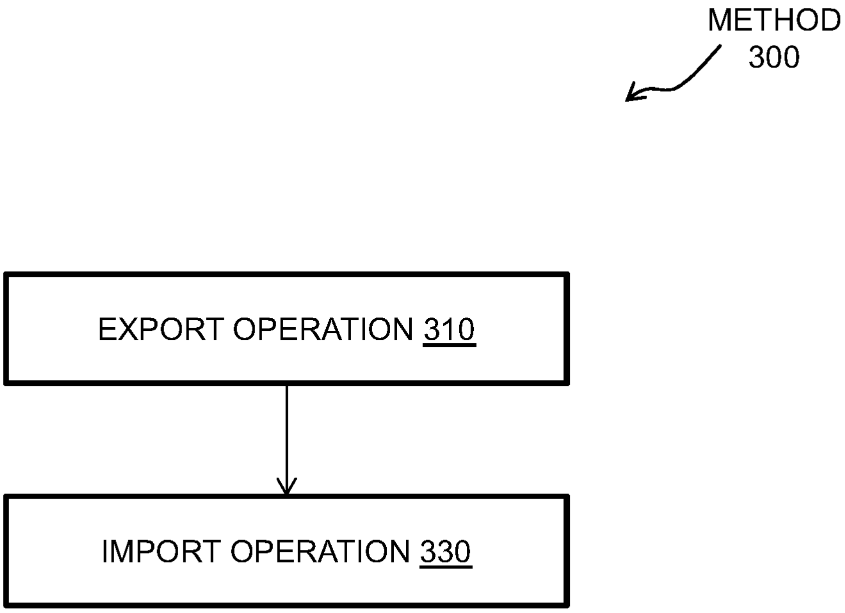


FIG. 3

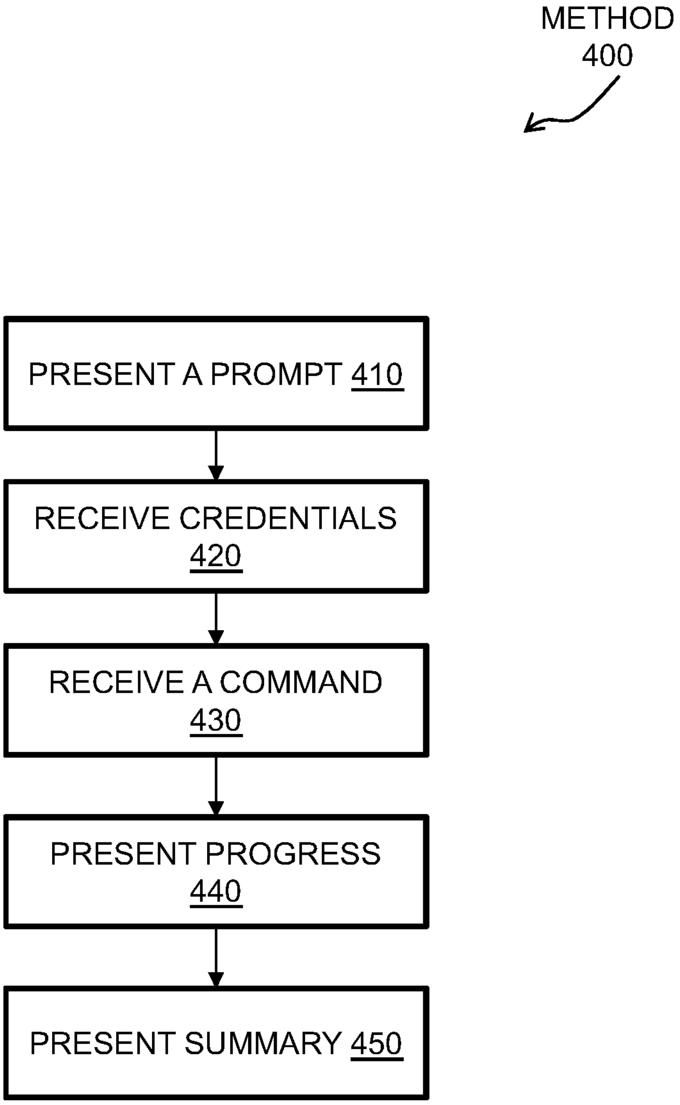

METHOD
400
PRESENT A PROMPT 410
RECEIVE CREDENTIALS 420
RECEIVE A COMMAND 430
PRESENT PROGRESS 440
PRESENT SUMMARY 450

FIG. 4

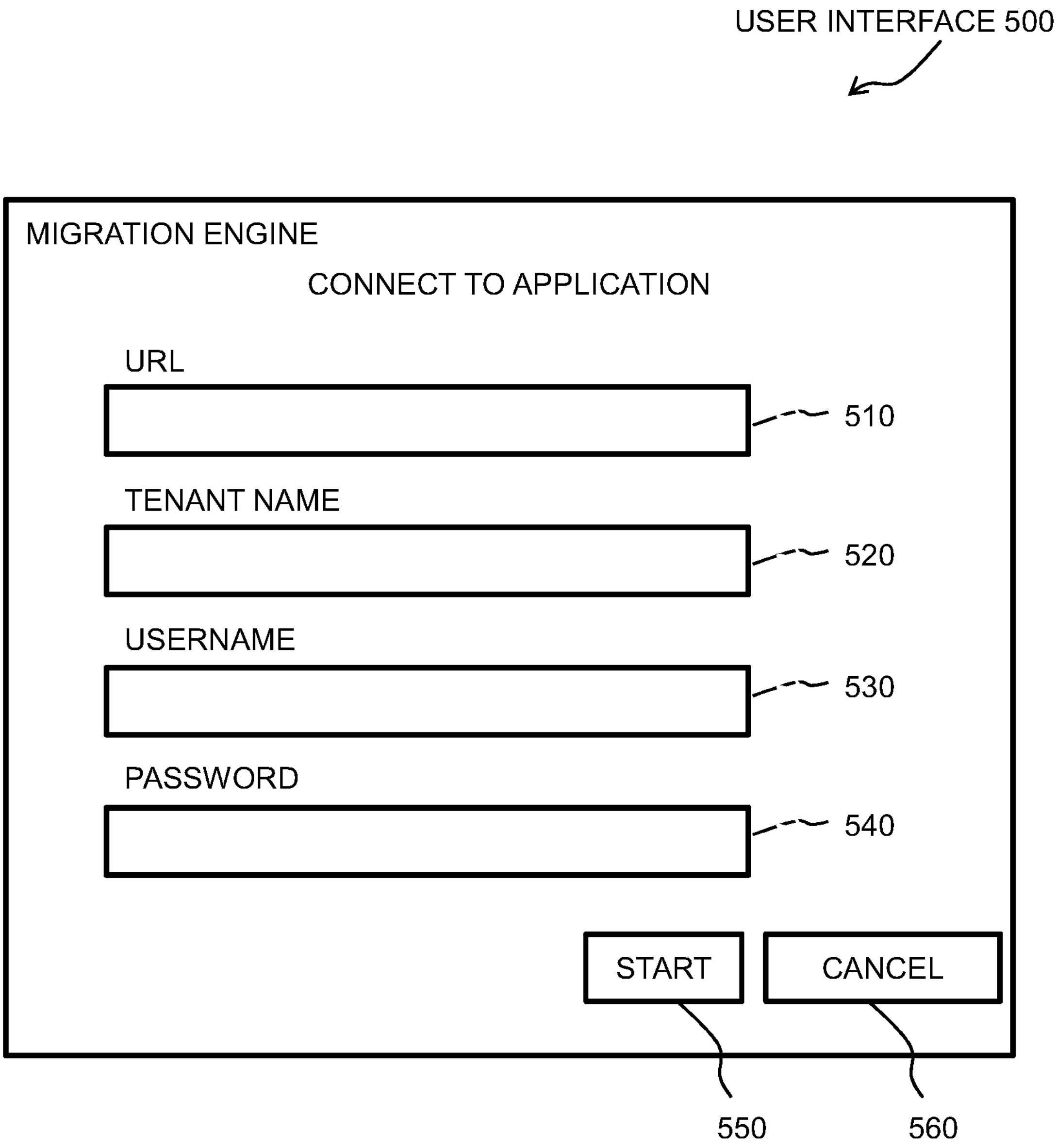
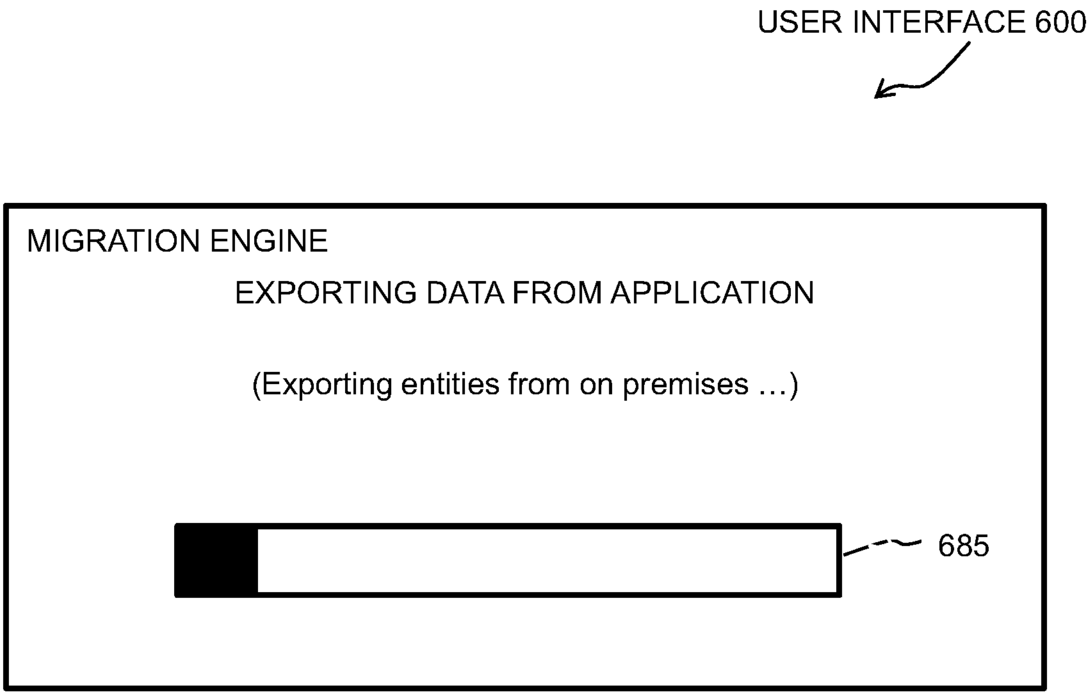

USER INTERFACE 500
MIGRATION ENGINE
CONNECT TO APPLICATION
URL
510
TENANT NAME
520
USERNAME
530
PASSWORD
540
START
CANCEL
550
560

FIG. 5

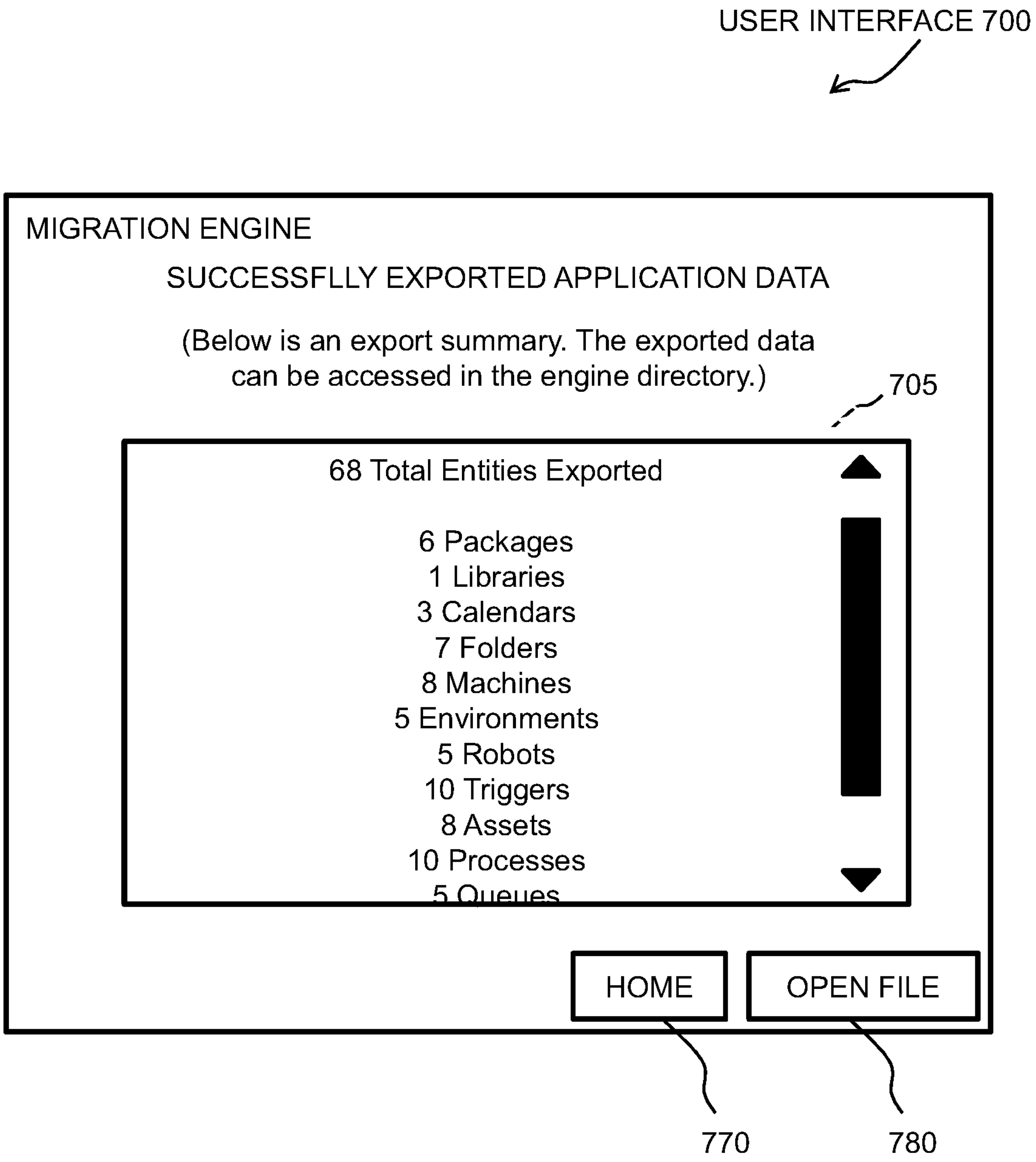


MIGRATION ENGINE

EXPORTING DATA FROM APPLICATION

(Exporting entities from on premises …)

FIG. 6



MIGRATION ENGINE

SUCCESSFLLY EXPORTED APPLICATION DATA

(Below is an export summary. The exported data can be accessed in the engine directory.)

68 Total Entities Exported

6 Packages
1 Libraries
3 Calendars
7 Folders
8 Machines
5 Environments
5 Robots
10 Triggers
8 Assets
10 Processes
5 Queues

FIG. 7

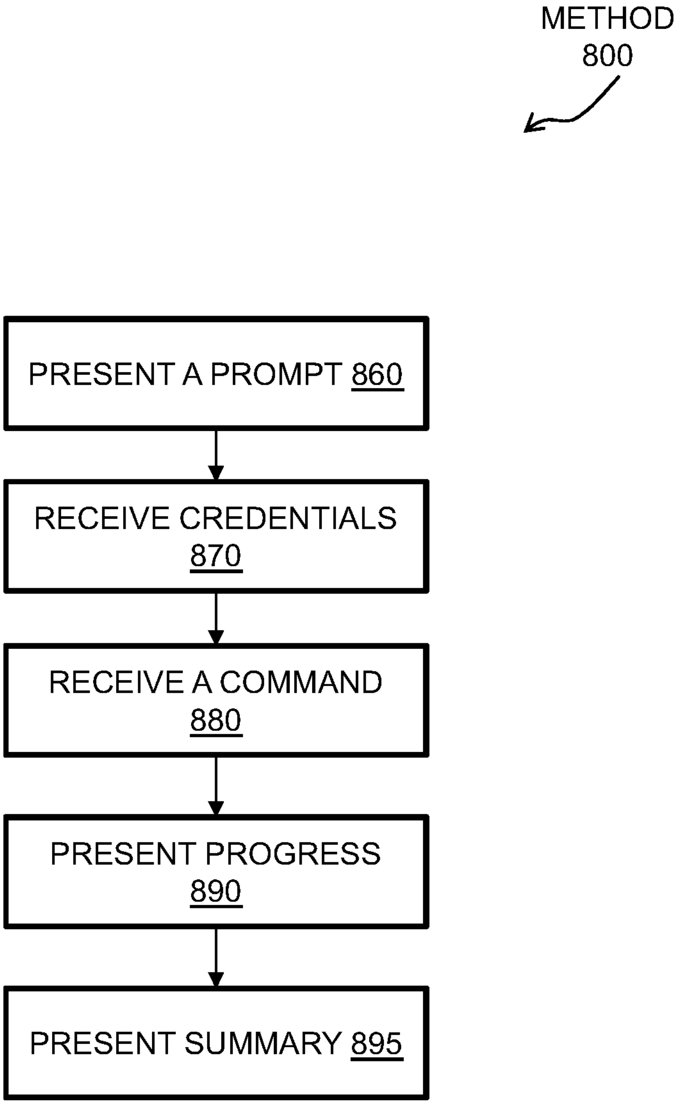

METHOD 800
PRESENT A PROMPT 860
RECEIVE CREDENTIALS 870
RECEIVE A COMMAND 880
PRESENT PROGRESS 890
PRESENT SUMMARY 895

FIG. 8

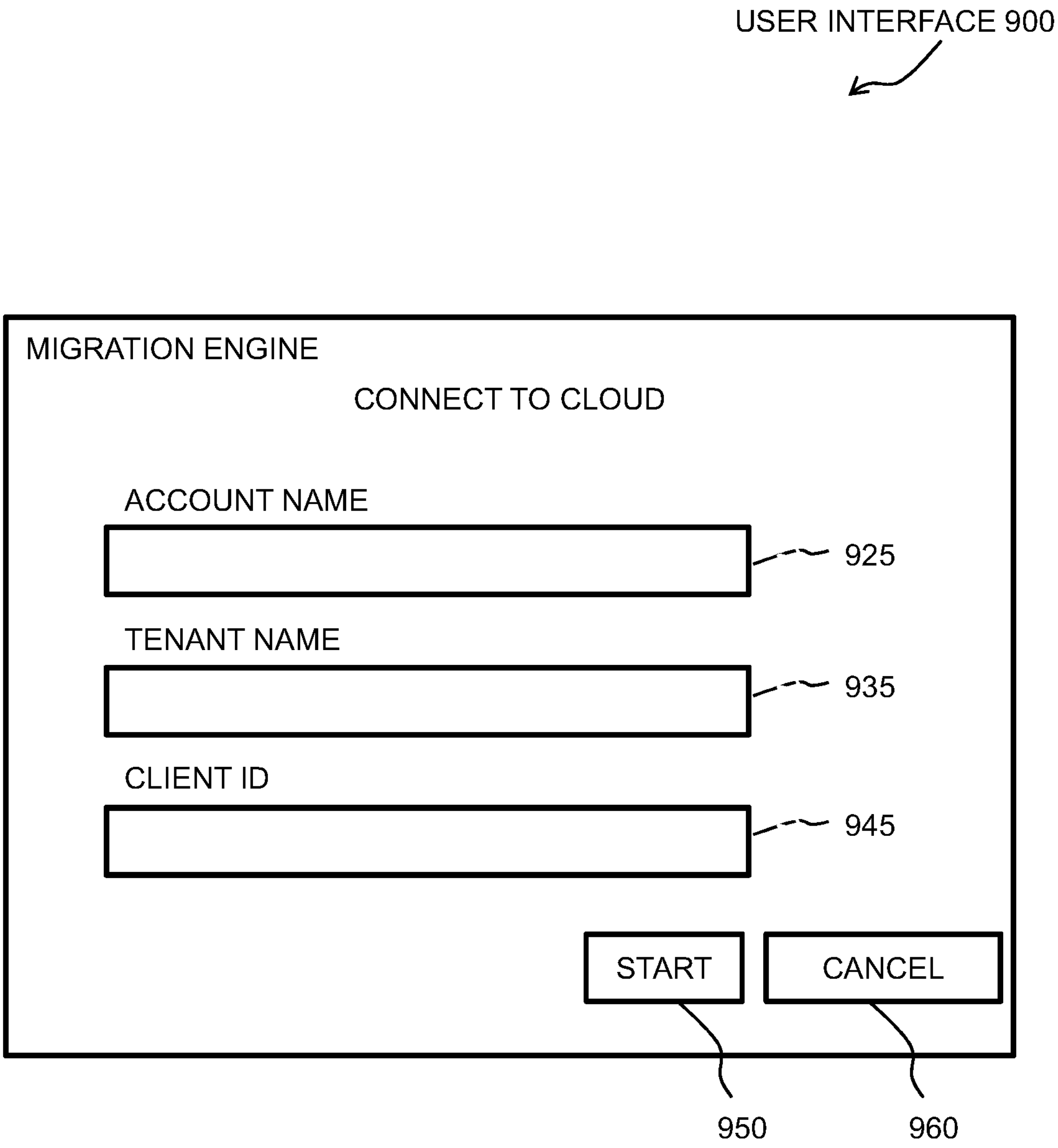
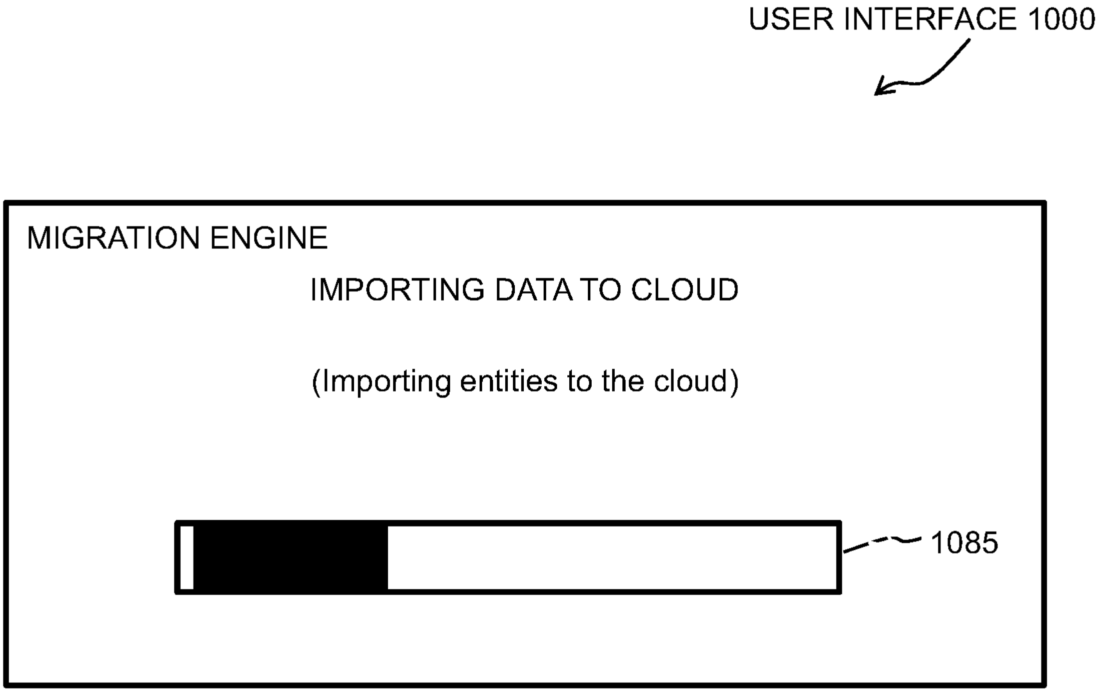

USER INTERFACE 900
MIGRATION ENGINE
CONNECT TO CLOUD
ACCOUNT NAME
925
TENANT NAME
935
CLIENT ID
945
START
CANCEL
950
960

FIG. 9

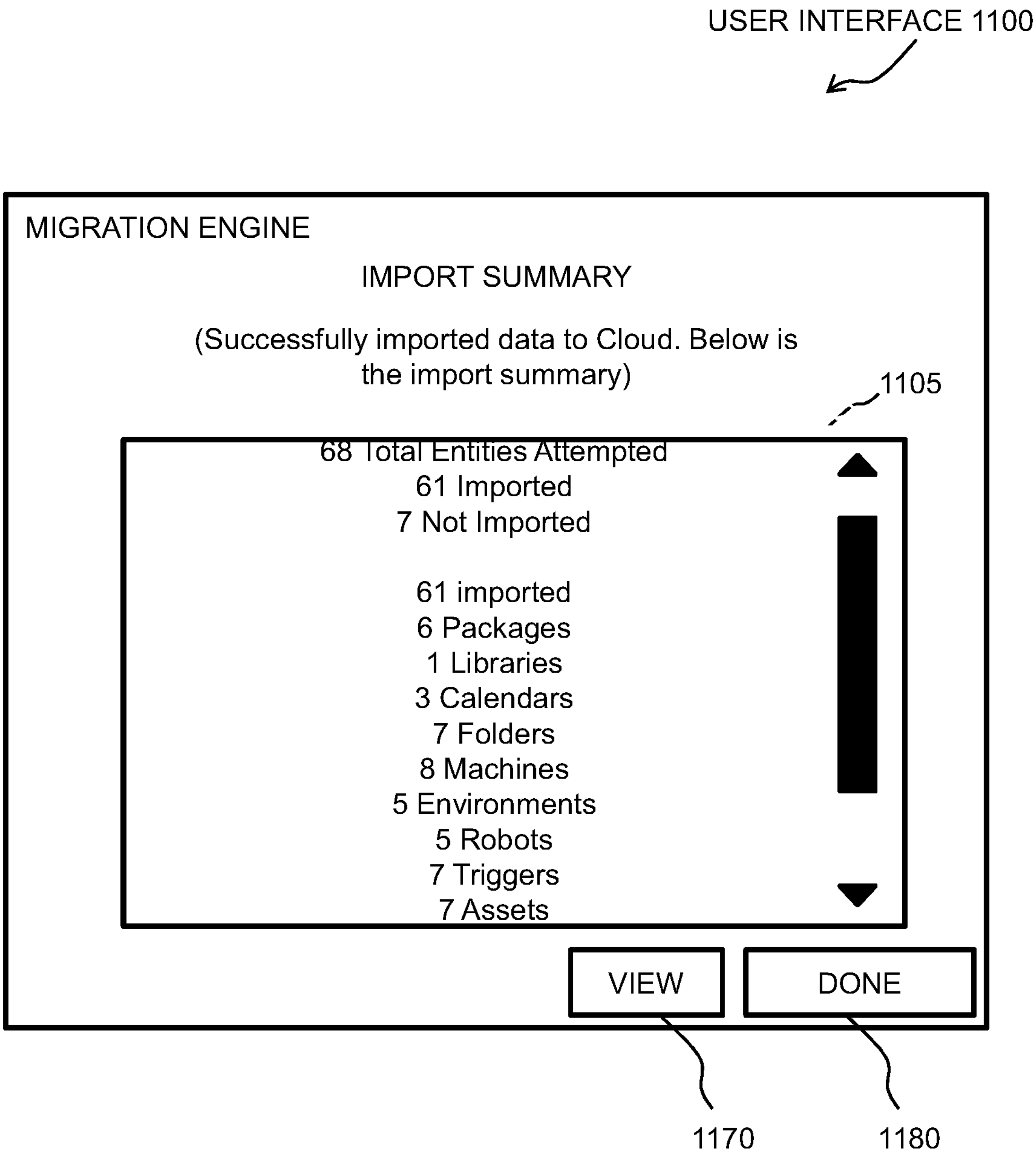


MIGRATION ENGINE

IMPORTING DATA TO CLOUD

(Importing entities to the cloud)

FIG. 10



IMPORT SUMMARY

(Successfully imported data to Cloud. Below is the import summary)

68 Total Entities Attempted
61 Imported
7 Not Imported

61 imported
6 Packages
1 Libraries
3 Calendars
7 Folders
8 Machines
5 Environments
5 Robots
7 Triggers
7 Assets

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2021/057184

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/455(2006.01)i; **G06F 9/48**(2006.01)i; **G06F 9/50**(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 9/455(2006.01); G05B 15/02(2006.01); G06F 16/11(2019.01); G06F 16/23(2019.01); G06F 17/30(2006.01); G06F 9/44(2006.01); G06F 9/445(2006.01); H04L 29/08(2006.01)

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: user data, automation, migration, premise, import, export, cloud

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2015-0277404 A1 (ROCKWELL AUTOMATION TECHNOLOGIES, INC.) 01 October 2015 (2015-10-01) Paragraphs [0004], [0038]-[0056], [0065], [0090]; and figures 2-4	1-20
Y	US 2020-0387476 A1 (MICROSOFT TECHNOLOGY LICENSING, LLC) 10 December 2020 (2020-12-10) Paragraphs [0032], [0041], [0079], [0099], [0106]; claim 11; and figures 1A-1B	1-20
A	US 2016-0357535 A1 (ORACLE INTERNATIONAL CORPORATION) 08 December 2016 (2016-12-08) Paragraphs [0023], [0064]-[0065]; claim 1; and figures 1, 9	1-20
A	US 2014-0279896 A1 (APPSENSE LIMITED) 18 September 2014 (2014-09-18) Paragraphs [0049]-[0052], [0086]; and figure 3	1-20
A	KR 10-2019-0060010 A (CONVIDA WIRELESS, LLC) 31 May 2019 (2019-05-31) Paragraphs [0121], [0132], [0256]; and figure 3	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 **16 June 2022**	Date of mailing of the international search report **16 June 2022**
Name and mailing address of the ISA/KR **Korean Intellectual Property Office** **189 Cheongsa-ro, Seo-gu, Daejeon** **35208, Republic of Korea** Facsimile No. **+82-42-481-8578**	Authorized officer **YANG, Jeong Rok** Telephone No. **+82-42-481-5709**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2021/057184

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2015-0277404 A1	01 October 2015	CN 104950836 A	30 September 2015
		EP 2924561 A2	30 September 2015
		EP 2924562 A2	30 September 2015
		EP 2924569 A2	30 September 2015
		EP 2924570 A2	30 September 2015
		EP 2924571 A2	30 September 2015
		EP 2924572 A2	30 September 2015
		EP 2924573 A2	30 September 2015
		EP 2924575 A2	30 September 2015
		US 10095202 B2	09 October 2018
		US 10208947 B2	19 February 2019
		US 10334048 B2	25 June 2019
		US 10510027 B2	17 December 2019
		US 2015-0276208 A1	01 October 2015
		US 2015-0277399 A1	01 October 2015
		US 2015-0277406 A1	01 October 2015
		US 2015-0281233 A1	01 October 2015
		US 2015-0281319 A1	01 October 2015
		US 2015-0281355 A1	01 October 2015
		US 2015-0281356 A1	01 October 2015
		US 2015-0281453 A1	01 October 2015
		US 2017-0154298 A1	01 June 2017
		US 2018-0063244 A1	01 March 2018
		US 2018-0300665 A1	18 October 2018
		US 9614963 B2	04 April 2017
		US 9825949 B2	21 November 2017
		US 9838476 B2	05 December 2017
		US 9843617 B2	12 December 2017
		US 9866635 B2	09 January 2018
		US 9886012 B2	06 February 2018
		US 9971317 B2	15 May 2018
		US 9990596 B2	05 June 2018
US 2020-0387476 A1	10 December 2020	US 10216744 B2	26 February 2019
		US 10846262 B2	24 November 2020
		US 2016-0321275 A1	03 November 2016
		US 2019-0197020 A1	27 June 2019
		WO 2016-178923 A1	10 November 2016
US 2016-0357535 A1	08 December 2016	US 10324697 B2	18 June 2019
US 2014-0279896 A1	18 September 2014	US 9465856 B2	11 October 2016
KR 10-2019-0060010 A	31 May 2019	CN 107079050 A	18 August 2017
		CN 112217905 A	12 January 2021
		EP 3195571 A1	26 July 2017
		EP 3913896 A2	24 November 2021
		JP 2017-535838 A	30 November 2017
		JP 2018-147507 A	20 September 2018
		JP 6335388 B2	30 May 2018
		JP 6563555 B2	21 August 2019
		KR 10-1984120 B1	03 September 2019
		KR 10-2017-0056013 A	22 May 2017

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2021/057184

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
		KR 10-2021-0002132 A	06 January 2021
		KR 10-2021-0124533 A	14 October 2021
		KR 10-2198229 B1	04 January 2021
		KR 10-2311627 B1	13 October 2021
		KR 10-2360767 B1	14 February 2022
		US 10367896 B2	30 July 2019
		US 11064033 B2	13 July 2021
		US 2017-0289271 A1	05 October 2017
		US 2019-0297149 A1	26 September 2019
		US 2021-0297493 A1	23 September 2021
		WO 2016-044718 A1	24 March 2016