



- (51) **International Patent Classification:**
G06F 9/50 (2006.01) G06F 11/07 (2006.01)
G06F 11/34 (2006.01)
- (21) **International Application Number:**
PCT/US2018/064153
- (22) **International Filing Date:**
06 December 2018 (06.12.2018)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
15/834,369 07 December 2017 (07.12.2017) US
- (71) **Applicant: ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, M/S 5op7, Redwood Shores, CA 94065 (US).
- (72) **Inventors: LODHA, Suryaveer, Singh;** 420 Mission Bay Blvd. N., Unit 706, San Francisco, CA 94158 (US). **BLA-BY, William;** 1100 Vista Point Lane, Concord, CA 94521 (US).

(74) **Agent: KRAGULJAC, Petar;** Kraguljac Law Group, LLC, 4700 Rockside Road, Summit One, Suite 510, Independence, OH 44131 (US).

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

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

(54) **Title:** METHOD AND SYSTEM FOR TRACKING APPLICATION ACTIVITY DATA FROM REMOTE DEVICES AND GENERATING A CORRECTIVE ACTION DATA STRUCTURE FOR THE REMOTE DEVICES

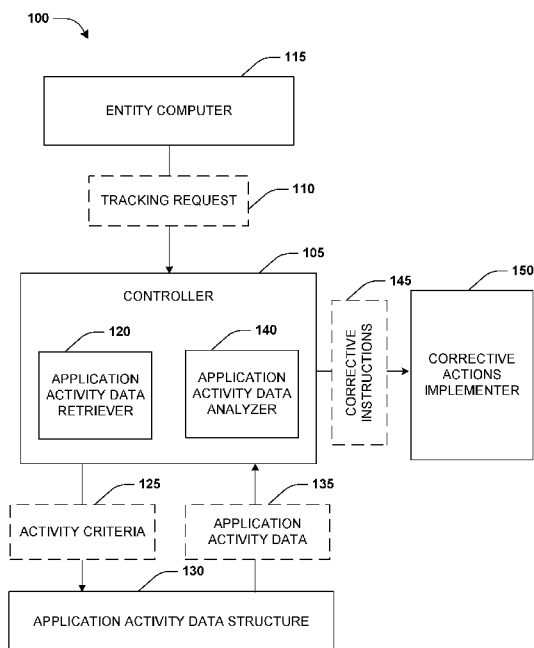


FIG. 1

(57) **Abstract:** Systems, methods, and other embodiments associated with generating a corrective action data structure for a set of remote devices based upon corrective actions are described. In one embodiment, a method includes receiving application activity data from a plurality of remote devices. The application activity data is analyzed to determine a set of application activities associated with each of the plurality of remote devices, and the sets of application activities are analyzed to determine one or more challenges associated with a set of remote devices of the plurality of remote devices. Corrective actions are determined for each of the one or more challenges, and a corrective action data structure is generated for the set of remote devices based upon the corrective actions.



Published:

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

**METHOD AND SYSTEM FOR TRACKING APPLICATION ACTIVITY
DATA FROM REMOTE DEVICES AND GENERATING A CORRECTIVE
ACTION DATA STRUCTURE FOR THE REMOTE DEVICES**

BACKGROUND

[0001] Computing devices are used to implement various services and products. For example, various computing devices may provide a scheduling service, such as scheduling of tasks to be performed using vehicles, and a communication service, such as transmitting messages to, and receiving messages from, other computing devices. Scheduled tasks and transmitted/received messages may be stored within databases or other storage structures of a distributed network environment (e.g., a cloud service), or within databases or other storage structures of a local computer. Users may interact with each of the computing devices to perform operations upon corresponding scheduled tasks and corresponding transmitted/received messages. For example, each computing device may be used to add new tasks, to remove existing tasks, or to modify existing tasks, or to generate new messages, remove existing messages, or modify existing messages.

[0002] The scheduling service and the communication service may have bugs or shortcomings that may hinder the usage of the services by users of some of the computing devices. Such bugs or shortcomings may not be identified or known to the providers of the services.

[0003] In order to improve the performance of the computing devices, the effectiveness of the services, the rate of successful completion of tasks and transmission of messages, and satisfaction of customers of the services, it is desirable for the providers of the services to be able to efficiently identify and correct bugs or shortcomings of the services.

[0004] Unfortunately, typical existing techniques are limited to using observational and anecdotal information to identify bugs or shortcomings. Thus, the improvement of services has been restricted by having a limited, and often slowly received, source of information.

5

SUMMARY

[0005] In one embodiment, a non-transitory computer-readable medium stores computer-executable instructions that, when executed by at least a processor of a computer, causes the computer to: receive, by at least the processor, application activity data from a plurality of remote devices; analyze, by at least the processor, the application activity data to determine a set of application activities associated with each of the plurality of remote devices; analyze, by at least the processor, the sets of application activities to determine one or more challenges associated with a set of remote devices of the plurality of remote devices; determine, by at least the processor, corrective actions for each of the one or more challenges; and generate, by at least the processor, a corrective action data structure for the set of remote devices based upon the corrective actions.

[0006] According to another embodiment, the corrective action data structure includes code that when received and executed by a remote device causes a modification to an application that mitigates at least one of the one or more challenges.

[0007] According to another embodiment, the corrective action data structure includes a report identifying (i) the one or more challenges associated with the set of remote devices, and (ii) the corrective actions determined for the one or more challenges.

[0008] According to another embodiment, the computer-executable instructions to analyze the sets of application activities to determine one or more challenges

associated with the set of remote devices of the plurality of remote devices further include computer-executable instructions that when executed by the processor cause the computer to: determine that a first set of application activities of a first remote device of the plurality of remote devices is associated with a first challenge;
5 determine that a second set of application activities of a second remote device of the plurality of remote devices is associated with the first challenge; and responsive to determining that the first remote device and the second remote device are both associated with the first challenge, generate the set of remote devices in association with the first challenge, wherein the set of remote devices
10 include the first remote device and the second remote device.

[0009] According to another embodiment, the computer-executable instructions further include instructions that, when executed by at least the processor, cause the computer to transmit the corrective action data structure to each remote device of the set of remote devices.

15 **[0010]** According to another embodiment, the computer-executable instructions further include instructions that, when executed by at least the processor, cause the computer to: generate a graphical user interface based upon the corrective action data structure; and provide the graphical user interface for display.

[0011] According to another embodiment, the computer-executable instructions
20 further include instructions that, when executed by at least the processor, cause the computer to transmit the corrective action data structure for the set of remote devices to a managing entity of the set of remote devices.

[0012] According to another embodiment, a computing system includes: a processor connected to memory; and a corrective module stored on a non-
25 transitory computer readable medium and configured with instructions that when executed by the processor cause the processor to: receive application activity data from a plurality of remote devices; analyze the application activity data to determine a set of application activities associated with each of the plurality of

remote devices; analyze the sets of application activities to determine one or more challenges associated with a set of remote devices of the plurality of remote devices; determine corrective actions for each of the one or more challenges; and generate a corrective action data structure for the set of remote devices based upon the corrective actions.

[0013] According to another embodiment, the corrective action data structure includes code that when received and executed by a remote device causes a modification to an application that mitigates at least one of the one or more challenges.

[0014] According to another embodiment, the corrective action data structure includes a report identifying (i) the one or more challenges associated with the set of remote devices, and (ii) the corrective actions determined for the one or more challenges.

[0015] According to another embodiment, the instructions to analyze the sets of application activities to determine one or more challenges associated with the set of remote devices of the plurality of remote devices further include instructions that when executed by the processor cause the processor to: determine that a first set of application activities of a first remote device of the plurality of remote devices is associated with a first challenge; determine that a second set of application activities of a second remote device of the plurality of remote devices is associated with the first challenge; and responsive to determining that the first remote device and the second remote device are both associated with the first challenge, generate the set of remote devices in association with the first challenge, wherein the set of remote devices include the first remote device and the second remote device.

[0016] According to another embodiment, the corrective module is further configured with instructions that when executed by the processor cause the

processor to transmit the corrective action data structure to each remote device of the set of remote devices.

[0017] According to another embodiment, the corrective module is further configured with instructions that when executed by the processor cause the processor to: generate a graphical user interface based upon the corrective action data structure; and provide the graphical user interface for display.

[0018] According to another embodiment, the corrective module is further configured with instructions that when executed by the processor cause the processor to transmit the corrective action data structure for the set of remote devices to a managing entity of the set of remote devices.

[0019] According to another embodiment, a computer-implemented method, which is performed by a computing device including a processor, includes: receiving, by at least the processor, application activity data from a plurality of remote devices; analyzing, by at least the processor, the application activity data to determine a set of application activities associated with each of the plurality of remote devices; analyzing, by at least the processor, the sets of application activities to determine one or more challenges associated with a set of remote devices of the plurality of remote devices; determining, by at least the processor, corrective actions for each of the one or more challenges; and generating, by at least the processor, a corrective action data structure for the set of remote devices based upon the corrective actions.

[0020] According to another embodiment, the corrective action data structure includes code that when received and executed by a remote device causes a modification to an application that mitigates at least one of the one or more challenges.

[0021] According to another embodiment, the corrective action data structure includes a report identifying (i) the one or more challenges associated with the set

of remote devices, and (ii) the corrective actions determined for the one or more challenges.

[0022] According to another embodiment, analyzing the sets of application activities to determine one or more challenges associated with the set of remote devices of the plurality of remote devices includes: determining that a first set of application activities of a first remote device of the plurality of remote devices is associated with a first challenge; determining that a second set of application activities of a second remote device of the plurality of remote devices is associated with the first challenge; and responsive to determining that the first remote device and the second remote device are both associated with the first challenge, generating the set of remote devices in association with the first challenge, wherein the set of remote devices include the first remote device and the second remote device.

[0023] According to another embodiment, the computer-implemented method further includes transmitting the corrective action data structure to each remote device of the set of remote devices.

[0024] According to another embodiment, the computer-implemented method further includes transmitting the corrective action data structure for the set of remote devices to a managing entity of the set of remote devices.

[0025] The above summary presents a simplified summary in order to provide a basic understanding of some aspects of the systems and/or methods discussed herein. This summary is not an extensive overview of the systems and/or methods discussed herein. It is not intended to identify key/critical elements or to delineate the scope of such systems and/or methods. Its sole purpose is to present some concepts in a simplified form as a prelude to the more detailed description that is presented later.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate various systems, methods, and other embodiments of the disclosure. It will be appreciated that the illustrated element boundaries (e.g., boxes, groups of boxes, or other shapes) in the figures represent one embodiment of the boundaries. In some embodiments one element may be implemented as multiple elements or that multiple elements may be implemented as one element. In some embodiments, an element shown as an internal component of another element may be implemented as an external component and vice versa. Furthermore, elements may not be drawn to scale.

[0027] FIG. 1 illustrates an embodiment of a system associated with determining corrective actions for challenges associated with a set of remote devices.

[0028] FIG. 2 illustrates an embodiment of a method associated with determining corrective actions for challenges associated with a set of remote devices.

[0029] FIG. 3 illustrates an embodiment of an application activity data structure.

[0030] FIG. 4 illustrates an embodiment of an entity computer and a graphical user interface.

[0031] FIG. 5 illustrates an embodiment of a corrective actions implementer with control instructions.

[0032] FIG. 6 illustrates an embodiment of a display screen and a graphical user interface.

[0033] FIG. 7 illustrates an embodiment of a non-transitory computer-readable medium.

[0034] FIG. 8 illustrates an embodiment of a computing system configured with the example systems and/or methods disclosed.

[0035] FIG. 9 illustrates an embodiment of a system associated with determining corrective actions for challenges associated with a set of remote devices.

DETAILED DESCRIPTION

[0036] Computerized systems and methods are described herein that provide for determining corrective actions for challenges associated with a set of remote devices. In one embodiment, a plurality of remote devices may each run one or more applications. For example, a first remote device may run a first application and a second application, and a second remote device may run (a different instance of) the first application and the second application.

[0037] The operation of each remote device may include performing various actions while attempting to achieve various objectives/results. Often, an initial action is not sufficient to achieve an objective, and additional actions are performed until the objective is reached. Thus, while operating the first application on the first remote device in order to meet a first objective/result, a first user may perform a first series of actions, or after performing a first action, may switch to the second application and perform a second action. Similarly, while operating the first application on the second remote device in order to meet the first objective/result, a second user may perform the first series of actions, or after performing the first action, may switch to the second application and perform the second action.

[0038] Unfortunately, users of such applications may face significant delay and hurdles to meeting their objectives/results, or may fail to achieve the objectives altogether. Such delays and failures can lead to a fall in satisfaction with the application and ultimately a decrease in use of the applications. It may be

appreciated that such delays and failures can occur even when the applications are technically capable of performing the desired objectives.

[0039] Thus, in order to optimize the performance of the applications, the actions performed on the various remote devices operating the applications may need to be analyzed to identify challenges and corrective actions that can address the identified challenges. By doing so, objectives may be reached using a reduced number of activities and resources, which may improve efficiency, resource (e.g., memory, bandwidth, processor, etc.) availability, and reduce power usage.

[0040] With reference to FIG. 1, one embodiment of a computerized system 100 associated with determining corrective actions for challenges associated with a set of remote devices is illustrated. The system 100 includes a controller 105, which may be configured to execute on a computer. The controller 105 may be initiated based upon a tracking request 110 being received from an entity computer 115, such as a computer of a manager of an application platform that is requesting that activities of one or more applications of the application platform on remote devices be tracked/monitored. For example, the application platform may provide various utility/productivity applications that are used for commercial purposes by various clients/companies. Such applications may include a customer self-service application, a customer care and billing application, a service order management application, a smart grid gateway application, a meter data management application, a work and asset management application, an operational device management application, a mobile workforce management application, a distribution management system application, or an outage management system application. The controller 105 may receive the tracking request 110 over a network connection.

[0041] The controller 105 utilizes an application activity data retriever 120 to generate activity criteria 125 (of the tracking request 110) that identify data representative of an experience/story of one or more users with one or more

applications on remote devices. The activity criteria 125 may specify one or more applications to be tracked in accordance with the tracking request 110, such as a customer care and billing application and a work and asset management application, or a type of activity/interaction, of a user with one or more applications, to be tracked. The type of activity/interaction may include a click, a keystroke, performance of an action, modification of a graphical rendering, or a selection of a user element, for example.

[0042] The activity criteria 125 are used to analyze an application activity data structure 130 in order to identify data records to be considered for the tracking request 110. The application activity data structure 130 includes data records of activities/interactions of users with applications. For example, data records indicative of clicks, actions, and modifications of a first application on a first remote device, selections of a second application on a second remote device, and actions of a third application on a third remote device may be aggregated and stored in the application activity data structure 130.

[0043] The controller 105 uses the application activity data structure 130 to generate application activity data 135 representative of the activities/interactions of users with applications that match the activity criteria 125. For example, a first activity criteria, such as the type of applications that the tracking request 110 is directed to tracking, and a second activity criteria, such as the type of activity to be tracked, may be used to search the application activity data structure 130. In some examples, the application activity data 135 may include representations of activities that meet at least one of the activity criteria 125, such as any activities that match at least one of the first activity criteria or the second activity criteria. In other examples, the application activity data 135 may include representations of activities that meet two or more, or all, of the activity criteria 125, such as activities that match both the first activity criteria and the second activity criteria (but not

activities that only match one of the first activity criteria or the second activity criteria).

[0044] The controller 105 utilizes the application activity data analyzer 140 to analyze the application activity data 135 to determine a set of application activities associated with each of the plurality of remote devices. For example, a first set of application activities associated with a first remote device may be identified in the application activity data 135, and a second set of application activities associated with a second remote device may be identified in the application activity data 135.

[0045] The sets of application activity data may be analyzed to determine one or more challenges associated with a set of remote devices. In some examples, each set of application activity data may be analyzed individually to determine a challenge. For example, analysis of the first set of application activities may be performed to determine a first challenge associated with the first remote device, while analysis of the second set of application activities may be performed to determine a second challenge associated with the second remote device. In some examples, a plurality of sets of application activity data may be analyzed together to determine a challenge. For example, analysis of the first set of application activities and the second set of application activities (in combination) may be performed to determine a third challenge associated with the first remote device and the second remote device.

[0046] In some examples, remote devices are grouped into sets based upon their association with common challenges. For example, the first remote device with the first set of application activities and the second remote device with the second set of application activities may be grouped together into an exemplary set of remote devices in response to determining that both remote devices (and their application activities) face the (common) third challenge. A different remote device that does not face the third challenge is not included in the exemplary set of remote devices, but may be included in a different set of remote devices generated in

association with a (common) fourth challenge. The application activities of each set of remote devices may be analyzed together to determine one or more challenges for the corresponding set of remote devices.

[0047] The controller 105 utilizes the application activity data analyzer 140 to determine corrective actions for each of the one or more challenges. For example, introduction of a new feature to an application may be determined to be the appropriate corrective action for the first challenge, while modification of an appearance of a graphical user interface element may be determined to be the appropriate corrective action for the second challenge, and removal of an old feature in an application may be determined to be the appropriate corrective action for the third challenge.

[0048] In some examples, the determination of corrective actions is based upon a matching of each identified challenge with a corresponding corrective action in a database. In some examples, the determination of corrective actions is based upon a comparison of value predicted to result from performing one or more corrective actions for each challenge. For example, a first value predicted to result from performing a first corrective action may be compared to a second value predicted to result from performing a second corrective action, and the corrective action associated with the higher value is selected.

[0049] The controller 105 generates corrective instructions 145 to provide to, and thus control, the corrective actions implementer 150. The corrective instructions 145 may provide for the corrective actions implementer 150 to perform the corrective actions corresponding to the challenges. The corrective instructions 145 includes a corrective action data structure comprising code that when transmitted to one or more remote devices running an application, causes a modification to the application that overcomes or mitigates at least one challenge. For example, the corrective instructions 145 may provide for transmitting an upgrade that adds the new feature to the first application, for transmitting a patch

that modifies the graphical user interface element in the second application, or for transmitting a patch that removes the old feature in the first application and the second application (or generates an intermediate application to connect the first application with the second application).

5 **[0050]** Code for a corrective action may be transmitted to the corresponding set of remote devices associated with the challenge corresponding to the corrective action. For example, the upgrade adding the new feature may be transmitted to a first set of remote devices, the patch that modified the graphical user interface element may be transmitted to a second set of remote devices, and the path that
10 removes the old feature may be transmitted to a third set of remote devices. In the example, the third set of remote devices, which includes remote devices running the first application and the second application, may include at least some of the remote devices of the first set of remote devices, which run the first application, and at least some of the remote devices of the second set of remote devices, which
15 run the second application.

[0051] In some examples, the corrective instructions 145 are transmitted to a managing entity of remote devices determined to face the corresponding challenges. The managing entity can include a service providing or operating the applications on a plurality of remote devices, such a creator or service provider of
20 the applications. The managing entity can analyze the corrective instructions 145 and generate code, based upon the analysis, to address the challenges. For example, the code generated may correspond to a patch or upgrade for an application that the managing entity may transmit to remote devices, or may correspond to a path, upgrade or fix for a framework used for one or more
25 applications.

[0052] In some examples, the corrective instructions 145 comprise a corrective action data structure that comprises a report. The report identifies the challenges identified in association with the set of remote devices, and identifies the corrective

actions determined to address the challenges. For example, the corrective instructions 145 may provide for the corrective actions implementer 150 to change a graphical representation of the sets of application activities associated with the challenges to visually distinguish the graphical representation of the sets of application activities associated with the challenges from graphical representations of other application activities (not associated with the challenges). The controlling of the corrective actions implementer 150 increases the speed at which the tracking request 110 is identified by reducing or eliminating time needed to search and navigate through various menus and displays, and thus improves the speed and usability of computer displays.

[0053] FIG. 9 illustrates one embodiment of a system 900 associated with determining corrective actions for challenges associated with a set of remote devices is illustrated, within which system 100 operates. Client 905 is a remote device that is tracked in accordance with the tracking request 110. Application activity data of the client 905 is transmitted to a web service 910 via a network connection. The web service 910 can be configured to run a cloud-native analytics service, which may collect and process data associated with web, mobile and event activities.

[0054] The web service 910 processes the activity data through various data structures such as an inbox 920, a processing data structure 925, an outbox 925, or an archive 930. A program 935 of the web service 910 may use activity data in the inbox 920 or the processing data structure 925 to generate a data set with a programming model, such as MapReduce. An Extract, Transform and Load (ETL) process pulls the data set and stores at least some of the data into the outbox 925. In some embodiments, the ETL process also pulls data from an asset data structure 940, and stores at least some of that data into the outbox 925. The program 935 diverts some of the activity data from the processing data structure 925 to the archive 930 based upon defined rules. Each portion of the activity data

is sorted based upon the applications that the portion corresponds to, and stored in the outbox 925 after being sorted.

[0055] A data center 915 retrieves the sorted activity data from the outbox 925 of the web service 910 via a network connection. A program 945 of the data center 5 915 stores the sorted activity data in a data warehouse 950 of the data center 915. In some embodiments, the program 945 modifies or formats the sorted activity data before storing the sorted activity data in the data warehouse 950.

[0056] In some embodiments, the entity computer 115 and the controller 105 are both components of the data center 915. In other embodiments, the entity 10 computer 115 is a component of the data center 915 while the controller 105 is a component of the web service 910. In some embodiments, the data warehouse 950 of the data center 915 includes the application activity data structure 130.

[0057] FIG. 2 illustrates one embodiment of a computer-implemented method 200 associated with determining corrective actions for challenges associated with 15 a set of remote devices. In one embodiment, method 200 is performed by the controller 105 utilizing various computing resources of the computer 805 (shown in Fig. 8), such as the processor 810 for executing instructions, memory 815 and/or disks 830 for storing data structures within which control instructions are generated, and/or network hardware for transmitting data structures to remote 20 computers over networks. The method 200 may be triggered based upon various triggers, such as receipt of the tracking request 110 from the entity computer 115, etc. In some examples, the tracking request 110 is received by a server (hosting the controller 105), from the entity computer 115.

[0058] Fig. 4 illustrates one embodiment of a graphical user interface 405 on 25 the entity computer 115 that may be used to generate the tracking request 110. The graphical user interface 405 is controlled to display a first graphical object 410 configured to receive user input that selects general information, such as a name or temporal scope, of the tracking request 110, a second graphical object 415

configured to receive user input that selects application criteria (call center management application) that the tracking request 110 pertains to, and a third graphical object 420 configured to receive user input that selects remote device criteria (all remote devices in the United States) of the tracking request 110. It may
5 be appreciated that the remote device criteria may correspond to a geographic location, a physical distance, a common network, or a common company.

[0059] The graphical user interface 405 is further controlled, based upon the user input received for the first graphical object 410, the second graphical object 415, and the third graphical object 420, to display a fourth graphical object 425
10 configured to display a fifth graphical object 430 and a sixth graphical object 435. The fifth graphical object 430 is configured to display one or more applications (or combinations of applications) that are applicable to the tracking request 110. In some examples, the fifth graphical object 430 is configured to receive user input that selects at least one application (or combination of applications) from amongst
15 a plurality of applications (or combinations of applications).

[0060] The sixth graphical object 435 is configured to display one or more remote devices (or categories thereof) that are applicable to the tracking request 110. In some examples, the sixth graphical object 435 is configured to receive user input that selects at least one remote device (or a category thereof) from
20 amongst a plurality of remote devices (or categories thereof).

[0061] Returning to Fig. 2, at 205, application activity data 135 is received from a plurality of remote devices. In some examples, the application activity data 135 is received by the server. Instructions, such as JavaScript instructions, may be injected into applications (such as client browsers) on the remote devices, and may
25 be used to collect the application activity data 135 from each of the remote devices. In some examples, a data processing pipeline, such as an extract, transform, load (ETL) pipeline, may be used to curate the activity data. In some examples, the application activity data 135 is aggregated into the application activity data

structure 130, and then transmitted from the application activity data structure 130 to the controller 105. The application activity data 135 received from various remote devices and applications may share a common format, or may have different formats and be converted into the common format before being stored in
5 the application activity data structure 130.

[0062] One embodiment of the application activity data structure 130 is illustrated in Fig. 3 and includes data records of a plurality of applications 305, a plurality of activities 310, devices 315 that the respective activities were performed upon, and timestamps 320 that indicate the time of performance of one or more of
10 the plurality of activities 310.

[0063] In some examples, activities that may be tracked and stored in the application activity data structure 130 include a number of users associated with each utility, a level of proficiency of the user (e.g., in using an application), a frequency of logging into a service, a length of time spent on a website or
15 application in each session on a remote device, the role of one or more modules used on a remote device, a frequency of accessing selected features (such as a dashboard, an export feature, or an explore feature) on a remote device, which tabs is accessed for the greatest amount of time on a remote device, an amount of time spent on selected features (filters, datasets, calculations, or scenarios) on
20 a remote device, a frequency of checking related point information (sibling information or parent information) on a remote device, which filter is used most commonly on a remote device, what is the percentage of usage of each filter type (simple, advanced, or algorithm), which panel is used on a remote device, which drop downs are used on a remote device, how frequently a list view is accessed
25 on a remote device, how often a panel is exported from a remote device, whether an exported panel is in list view or graph view, what information a remote device displays on a map in addition to default settings (individual points, pop-ups, or neighboring point identifiers), the type of information that is most commonly

exported, how frequently data is viewed on a remote device but not exported, a format that is used to export data, which drill-down panel is used most commonly on a remote device, for which panel in summary view are changes made on a remote device to view different information, which information is selected on a remote device, or how often a link is used on a remote device.

[0064] Returning to Fig. 2, at 210, the application activity data 135 is analyzed to determine a set of application activities associated with each of the plurality of remote devices. Each set of application activities may be determined by aggregating application activities associated with a common web browser, IP address, cloud provider, or customer account. The application activity data 135 may be scanned to identify patterns associated with bots or scrapers, and activity data associated with the identified patterns may be removed to eliminate noise in the data and improve the accuracy of analysis. In some examples, the application activities of multiple remote devices can be aggregated into a single set of application activities, based upon a determination that the multiple remote devices share a common characteristic, such as running a same application, or a same combination of applications. In some examples, the sets of application activities represent the experiences/stories of one or more users with one or more applications on remote devices, and are retrieved via a network communication.

[0065] At 215, the sets of application activities are analyzed to determine one or more challenges associated with a set of remote devices of the plurality of remote devices. In some examples, a first challenge (or set of challenges) may be determined based upon a first combination of one or more activities performed on a first user device via a first application. For example, based upon the data records in Fig. 3 indicating that device (3) viewed an account tree in a call center manager application and opened a branch in the account tree in the call center manager application, a determination may be made that the user of device (3) is likely facing a first challenge and/or attempting to perform a first objective/result.

[0066] In some examples, a second challenge (or set of challenges) may be determined based upon a second combination of one or more activities performed on the first user device on the first application and a second application. For example, based upon the data records in Fig. 3 indicating that device (3) viewed
5 an account tree in a call center manager application, opened a branch in the account tree in the call center manager application, and browsed a customer profile in a customer billing application, a determination may be made that the user of device (3) is likely facing the second challenge and/or attempting to perform a second objective/result.

10 **[0067]** In some examples, the sets of application activities are analyzed to determine a probability of one or more challenges being associated with the set of remote devices. One or more challenges that are associated with a probability that exceeds a threshold may be selected, or the challenges may be ranked based upon their respective probabilities, and one or more of the highest ranking
15 challenges may be selected.

[0068] In some examples, a challenge is determined to be associated with a remote device (or a set of remote devices) in response to determining that the application activities indicative of the challenge were identified in a threshold number of remote devices. For example, the first challenge may be determined to
20 be associated with the first remote device (or a first set of remote devices) in response to determining that at least five remote devices performed the activities (or similar activities) that correspond to the first challenge.

[0069] The threshold number of remote devices may apply to every challenge, or may vary (and be dynamically determined) based upon a type of a challenge.
25 For example, challenges that correspond to suspected graphical user interface shortcomings may have lower thresholds than challenges that correspond to suspected lack of features.

[0070] In some examples, the threshold number of remote devices for each challenge may vary (and be dynamically determined) based upon the type of corrective actions associated with the challenge. For example, challenges that are associated with corrective actions that merely require the installation of a patch
5 may have lower thresholds than challenges that require a complete upgrade or reinstall. In another example, the threshold number of remote devices may be dynamically determined for a challenge based upon an amount of code that, as a result of a corrective action of the challenge, is expected to be modified, added or removed.

[0071] Returning to Fig. 2, at 220, corrective actions are determined for each of the one or more challenges. The corrective actions may include a modification of code, such as a patch or an upgrade. In some examples, the corrective actions may include generating a report illustrating the challenges and the corresponding activities. The report may be specific to a set of remote devices that fit within
15 defined/requested demographics, or may be specific to a plurality of users that performed common/similar actions or share common activity-based characteristics. In some examples, the corrective actions may include generating corrective guidance/feedback that explains the challenge and steps that can be taken on the remote devices (such as by users) to avert the challenge.

[0072] At 225, a corrective action data structure is generated for the set of remote devices based upon the corrective actions. In some examples, the corrective actions are analyzed to determine a probability of each of the corrective actions addressing the corresponding challenges. One or more corrective actions that are associated with a probability that exceeds a threshold may be selected for
25 inclusion in the corrective action data structure, or the corrective actions may be ranked based upon their respective probabilities, and one or more of the highest ranking corrective actions may be selected. For example, where three corrective actions are determined for the first challenge, the corrective action of the three that

is determined to be most likely to address the first challenge and/or achieve a desired objective/result may be selected while the other corrective actions may be discarded.

[0073] The corrective action data structure can thus include code that is transmitted to one or more remote devices, and when executed, cause the remote devices to modify existing application code in a manner that overcomes the challenges. In some examples, the code can be transmitted to a server or manager of applications of the remote devices, which may re-transmit the code (or customize the code for each remote device, then transmit the customized code) to each remote device, or may modify framework code in the server.

[0074] In some examples, the corrective action data structure can include a report that is transmitted to the server or the manager of the applications, or can include corrective guidance that can explain (in step by step instructions) actions to be performed on each of the remote devices to overcome the corresponding challenges. The report can illustrate a journey of activity across one or more products by a remote device at an individual level, or multiple journeys at an aggregate level.

[0075] The corrective action data structure can be used to generate the corrective instructions 145. One embodiment of the corrective instructions 145 is illustrated in Fig. 5 and includes a first instruction 510 to transmit first code to device (3), a second instruction 515 to transmit a second report to a managing device associated with device (2), and a third instruction 520 to transmit corrective guidance/feedback to device (1). The corrective guidance/feedback may be generated for display on a display screen 600 the device (1).

[0076] One embodiment of the controlling of the display screen 600 is illustrated in Fig. 6. The display screen 600 renders a guidance graphical user interface 605, which includes a graphical object 610 graphically representing the activities on the device (1) that were detected and used to generate the corrective

guidance/feedback. The guidance graphical user interface 605 further includes a graphical object 615 graphically representing the objectives/results that is determined to be sought/attempted based upon the activities of the device (1). The guidance graphical user interface 605 further includes a graphical object 620 graphically representing the challenges that seem to be inhibiting the device (1) from reaching the objectives/results. The guidance graphical user interface 605 further includes a graphical object 625 graphically representing guidance/feedback, such as actions that can be taken on the device (1) in order to overcome the challenges and reach the objectives/results. The controlling of the display screen 600 increases the speed at which graphical objects associated with the objectives 615 are identified, and thus improves the speed and usability of computer displays.

[0077] In some examples, application activities may not be analyzed to determine challenges. Application activities may be analyzed to determine common activities across various remote devices, and a report may be generated identifying the number or proportion of remote devices that performed each activity, and demographic or technical identifiers of the remote devices.

[0078] In some examples, generation of a corrective action data structure may help an entity, such as a utility entity, understand how applications are being used by remote devices. Such generation may further support data driven product development. For example, based upon the corrective action data structure, the entity may prioritize bug fixes and features based on actual usage patterns, may determine which applications (and which versions of applications) warrant continued support, and determine whether new features are actually used by clients. Analytics may be provided to the entity illustrating how sets of applications are impacting their engagement with their customer base, providing metrics and actionable insights on improving their customer satisfaction and engagement numbers based upon the sets of applications.

[0079] Fig. 7 is an illustration of a scenario 700 involving an example non-transitory computer-readable medium 705. In one embodiment, one or more of the components described herein are configured as program modules, such as the controller 105, stored in the non-transitory computer-readable medium 705. The program modules are configured with stored instructions, such as processor-executable instructions 710, that when executed by at least a processor, such as processor 715, cause the computing device to perform the corresponding function(s) as described herein. For example, functionality of the controller 105, stored in the non-transitory computer-readable medium 705, may be executed by the processor 715 as the processor-executable instructions 710 to perform an embodiment 740 of the method 200 of Fig. 2.

[0080] The non-transitory machine readable medium 705 includes the processor-executable instructions 710 that when executed by a processor 715 cause performance of at least some of the provisions herein. The non-transitory machine readable medium 705 includes a memory semiconductor (e.g., a semiconductor utilizing static random access memory (SRAM), dynamic random access memory (DRAM), and/or synchronous dynamic random access memory (SDRAM) technologies), a platter of a hard disk drive, a flash memory device, or a magnetic or optical disc (such as a compact disk (CD), a digital versatile disk (DVD), or floppy disk). The example non-transitory machine readable medium 705 stores computer-readable data 720 that, when subjected to reading 725 by a reader 730 of a device 735 (e.g., a read head of a hard disk drive, or a read operation invoked on a solid-state storage device), express the processor-executable instructions 710. In some embodiments, the processor-executable instructions 710, when executed cause performance of operations, such as at least some of the example method 200 of Fig. 2, for example. In some embodiments, the processor-executable instructions 710 are configured to cause implementation

of a system, such as at least some of the example system 100 of Fig. 1, for example.

[0081] FIG. 8 illustrates a scenario 800 of an example computing device that is configured and/or programmed with one or more of the example systems and methods described herein, and/or equivalents. The example computing device
5 may be a computer 805 that includes a processor 810, a memory 815, and input/output ports 820 operably connected by a bus 825. In one example, the computer 805 may include logic of the controller 105 configured to facilitate the system 100 and/or the method 200 shown in Figs. 1 and 2. In different examples,
10 the logic of the controller 105 may be implemented in hardware, a non-transitory computer-readable medium 705 with stored instructions, firmware, and/or combinations thereof. While the logic of the controller 105 is illustrated as a hardware component attached to the bus 825, it is to be appreciated that in other embodiments, the logic of the controller 105 could be implemented in the
15 processor 810, stored in memory 815, or stored in disk 830.

[0082] In one embodiment, logic of the controller 105 or the computer 805 is a means (e.g., structure: hardware, non-transitory computer-readable medium, firmware) for performing the actions described. In some embodiments, the computing device may be a server operating in a cloud computing system, a server
20 configured in a Software as a Service (SaaS) architecture, a smart phone, laptop, tablet computing device, and so on.

[0083] The means may be implemented, for example, as an application specific integrated circuit (ASIC) programmed to implement rule based source sequencing for allocation. The means may also be implemented as stored computer
25 executable instructions that are presented to computer 805 as data 845 that are temporarily stored in memory 815 and then executed by processor 810.

[0084] The logic of the controller 105 may also provide means (e.g., hardware, non-transitory computer-readable medium 705 that stores executable instructions, firmware) for performing rule based source sequencing for allocation.

[0085] Generally describing an example configuration of the computer 805, the processor 810 may be a variety of various processors including dual
5 microprocessor and other multi-processor architectures. The memory 815 may include volatile memory and/or non-volatile memory. Non-volatile memory may include, for example, read-only memory (ROM), programmable read-only memory (PROM), and so on. Volatile memory may include, for example, random access
10 memory (RAM), static random-access memory (SRAM), dynamic random access memory (DRAM), and so on.

[0086] The disks 830 may be operably connected to the computer 805 via, for example, an input/output (I/O) interface (e.g., card, device) 835 and an input/output port 820. The disks 830 may be, for example, a magnetic disk drive, a solid state
15 disk drive, a floppy disk drive, a tape drive, a Zip drive, a flash memory card, a memory stick, and so on. Furthermore, the disks 830 may be a CD-ROM drive, a CD-R drive, a CD-RW drive, a DVD ROM, and so on. The memory 815 can store a process 840 and/or a data 845, for example. The disk 830 and/or the memory 815 can store an operating system that controls and allocates resources of the
20 computer 805.

[0087] The computer 805 may interact with input/output (I/O) devices via the I/O interfaces 835 and the input/output ports 820. Input/output devices may be, for example, a keyboard, a microphone, a pointing and selection device, cameras, video cards, displays, the disks 830, the network devices 850, and so on. The
25 input/output ports 820 may include, for example, serial ports, parallel ports, and USB ports. I/O controllers 855 may connect the I/O interfaces 835 to the bus 825.

[0088] The computer 805 can operate in a network environment and thus may be connected to the network devices 850 via the I/O interfaces 835, and/or the I/O

ports 820. Through the network devices 850, the computer 805 may interact with a network. Through the network, the computer 805 may be logically connected to remote computers. Networks with which the computer 805 may interact include, but are not limited to, a local area network (LAN), a new area network (WAN), and other networks.

[0089] In another embodiment, the described methods and/or their equivalents may be implemented with computer executable instructions. Thus, in one embodiment, a non-transitory computer readable/storage medium is configured with stored computer executable instructions of an algorithm/executable application that when executed by a machine(s) cause the machine(s) (and/or associated components) to perform the method. Example machines include but are not limited to a processor, a computer, a server operating in a cloud computing system, a server configured in a Software as a Service (SaaS) architecture, a smart phone, and so on). In one embodiment, a computing device is implemented with one or more executable algorithms that are configured to perform any of the disclosed methods.

[0090] In one or more embodiments, the disclosed methods or their equivalents are performed by either: computer hardware configured to perform the method; or computer instructions embodied in a module stored in a non-transitory computer-readable medium where the instructions are configured as an executable algorithm configured to perform the method when executed by at least a processor of a computing device.

[0091] While for purposes of simplicity of explanation, the illustrated methodologies in the figures are shown and described as a series of blocks of an algorithm, it is to be appreciated that the methodologies are not limited by the order of the blocks. Some blocks can occur in different orders and/or concurrently with other blocks from that shown and described. Moreover, less than all the illustrated blocks may be used to implement an example methodology. Blocks may be

combined or separated into multiple actions/components. Furthermore, additional and/or alternative methodologies can employ additional actions that are not illustrated in blocks.

[0092] The following includes definitions of selected terms employed herein.

5 The definitions include various examples and/or forms of components that fall within the scope of a term and that may be used for implementation. The examples are not intended to be limiting. Both singular and plural forms of terms may be within the definitions.

[0093] References to “one embodiment”, “an embodiment”, “one example”, “an

10 example”, and so on, indicate that the embodiment(s) or example(s) so described may include a particular feature, structure, characteristic, property, element, or limitation, but that not every embodiment or example necessarily includes that particular feature, structure, characteristic, property, element or limitation. Furthermore, repeated use of the phrase “in one embodiment” does not
15 necessarily refer to the same embodiment, though it may.

[0094] A “data structure”, as used herein, is an organization of data in a computing system that is stored in a memory, a storage device, or other computerized system. A data structure may be any one of, for example, a data field, a data file, a data array, a data record, a database, a data table, a graph, a
20 tree, a linked list, and so on. A data structure may be formed from and contain many other data structures (e.g., a database includes many data records). Other examples of data structures are possible as well, in accordance with other embodiments.

[0095] “Computer-readable medium” or “computer storage medium”, as used

25 herein, refers to a non-transitory medium that stores instructions and/or data configured to perform one or more of the disclosed functions when executed. Data may function as instructions in some embodiments. A computer-readable medium may take forms, including, but not limited to, non-volatile media, and volatile media.

Non-volatile media may include, for example, optical disks, magnetic disks, and so on. Volatile media may include, for example, semiconductor memories, dynamic memory, and so on. Common forms of a computer-readable medium may include, but are not limited to, a floppy disk, a flexible disk, a hard disk, a magnetic tape, other magnetic medium, an application specific integrated circuit (ASIC), a programmable logic device, a compact disk (CD), other optical medium, a random access memory (RAM), a read only memory (ROM), a memory chip or card, a memory stick, solid state storage device (SSD), flash drive, and other media from which a computer, a processor or other electronic device can function with. Each type of media, if selected for implementation in one embodiment, may include stored instructions of an algorithm configured to perform one or more of the disclosed and/or claimed functions.

[0096] “Logic”, as used herein, represents a component that is implemented with computer or electrical hardware, a non-transitory medium with stored instructions of an executable application or program module, and/or combinations of these to perform any of the functions or actions as disclosed herein, and/or to cause a function or action from another logic, method, and/or system to be performed as disclosed herein. Equivalent logic may include firmware, a microprocessor programmed with an algorithm, a discrete logic (e.g., ASIC), at least one circuit, an analog circuit, a digital circuit, a programmed logic device, a memory device containing instructions of an algorithm, and so on, any of which may be configured to perform one or more of the disclosed functions. In one embodiment, logic may include one or more gates, combinations of gates, or other circuit components configured to perform one or more of the disclosed functions. Where multiple logics are described, it may be possible to incorporate the multiple logics into one logic. Similarly, where a single logic is described, it may be possible to distribute that single logic between multiple logics. In one embodiment, one or more of these logics are corresponding structure associated with performing the

disclosed and/or claimed functions. Choice of which type of logic to implement may be based on desired system conditions or specifications. For example, if greater speed is a consideration, then hardware would be selected to implement functions. If a lower cost is a consideration, then stored instructions/executable application would be selected to implement the functions.

[0097] An “operable connection”, or a connection by which entities are “operably connected”, is one in which signals, physical communications, and/or logical communications may be sent and/or received. An operable connection may include a physical interface, an electrical interface, and/or a data interface. An operable connection may include differing combinations of interfaces and/or connections sufficient to allow operable control. For example, two entities can be operably connected to communicate signals to each other directly or through one or more intermediate entities (e.g., processor, operating system, logic, non-transitory computer-readable medium). Logical and/or physical communication channels can be used to create an operable connection.

[0098] “User”, as used herein, includes but is not limited to one or more persons, computers or other devices, or combinations of these.

[0099] While the disclosed embodiments have been illustrated and described in considerable detail, it is not the intention to restrict or in any way limit the scope of the appended claims to such detail. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the various aspects of the subject matter. Therefore, the disclosure is not limited to the specific details or the illustrative examples shown and described. Thus, this disclosure is intended to embrace alterations, modifications, and variations that fall within the scope of the appended claims.

[0100] To the extent that the term “includes” or “including” is employed in the detailed description or the claims, it is intended to be inclusive in a manner similar

to the term “comprising” as that term is interpreted when employed as a transitional word in a claim.

[0101] To the extent that the term “or” is used in the detailed description or claims (e.g., A or B) it is intended to mean “A or B or both”. When the applicants
5 intend to indicate “only A or B but not both” then the phrase “only A or B but not both” will be used. Thus, use of the term “or” herein is the inclusive, and not the exclusive use.

Claims

What is claimed is:

1. A non-transitory computer-readable medium storing computer-executable instructions that when executed by at least a processor of a computer cause the computer to:

- receive, by at least the processor, application activity data from a plurality of remote devices;
- analyze, by at least the processor, the application activity data to determine a set of application activities associated with each of the plurality of remote devices;
- analyze, by at least the processor, the sets of application activities to determine one or more challenges associated with a set of remote devices of the plurality of remote devices;
- determine, by at least the processor, corrective actions for each of the one or more challenges; and
- generate, by at least the processor, a corrective action data structure for the set of remote devices based upon the corrective actions.

2. The non-transitory computer-readable medium of claim 1, wherein the computer-executable instructions to analyze the sets of application activities to determine one or more challenges associated with the set of remote devices of the plurality of remote devices further include computer-executable instructions that when executed by the processor cause the computer to:

- determine that a first set of application activities of a first remote device of the plurality of remote devices is associated with a first challenge;

determine that a second set of application activities of a second remote device of the plurality of remote devices is associated with the first challenge; and

responsive to determining that the first remote device and the second remote device are both associated with the first challenge, generate the set of remote devices in association with the first challenge, wherein the set of remote devices include the first remote device and the second remote device.

3. The non-transitory computer-readable medium of claim 1, further comprising instructions that when executed by at least the processor cause the computer to transmit the corrective action data structure to each remote device of the set of remote devices.

4 The non-transitory computer-readable medium of claim 1, further comprising instructions that when executed by at least the processor cause the computer to:

generate a graphical user interface based upon the corrective action data structure; and

provide the graphical user interface for display.

5. The non-transitory computer-readable medium of claim 1, further comprising instructions that when executed by at least the processor cause the computer to transmit the corrective action data structure for the set of remote devices to a managing entity of the set of remote devices.

6. A computing system, comprising:
a processor connected to memory; and

a corrective module stored on a non-transitory computer readable medium and configured with instructions that when executed by the processor cause the processor to:

- receive application activity data from a plurality of remote devices;
- analyze the application activity data to determine a set of application activities associated with each of the plurality of remote devices;
- analyze the sets of application activities to determine one or more challenges associated with a set of remote devices of the plurality of remote devices;
- determine corrective actions for each of the one or more challenges; and
- generate a corrective action data structure for the set of remote devices based upon the corrective actions.

7. The computing system of claim 6, wherein the instructions to analyze the sets of application activities to determine one or more challenges associated with the set of remote devices of the plurality of remote devices further include instructions that when executed by the processor cause the processor to:
- determine that a first set of application activities of a first remote device of the plurality of remote devices is associated with a first challenge;
 - determine that a second set of application activities of a second remote device of the plurality of remote devices is associated with the first challenge; and
- responsive to determining that the first remote device and the second remote device are both associated with the first challenge, generate the set of remote devices in association with the first challenge, wherein the set of remote devices include the first remote device and the second remote device.

8. The computing system of claim 6, wherein the corrective module is further configured with instructions that when executed by the processor cause the processor to:

generate a graphical user interface based upon the corrective action data structure; and
provide the graphical user interface for display.

9. The computing system of claim 6, wherein the corrective module is further configured with instructions that when executed by the processor cause the processor to transmit the corrective action data structure for the set of remote devices to a managing entity of the set of remote devices.

10. A computer-implemented method performed by a computing device comprising a processor, the computer-implemented method comprising:
receiving, by at least the processor, application activity data from a plurality of remote devices;
analyzing, by at least the processor, the application activity data to determine a set of application activities associated with each of the plurality of remote devices;
analyzing, by at least the processor, the sets of application activities to determine one or more challenges associated with a set of remote devices of the plurality of remote devices;
determining, by at least the processor, corrective actions for each of the one or more challenges; and
generating, by at least the processor, a corrective action data structure for the set of remote devices based upon the corrective actions.

11. The computer-implemented method of claim 1, 6 or 10, wherein the corrective action data structure comprises code that when received and executed

by a remote device causes a modification to an application that mitigates at least one of the one or more challenges.

12. The computer-implemented method of claim 1, 6 or 10, wherein the corrective action data structure comprises a report identifying (i) the one or more challenges associated with the set of remote devices, and (ii) the corrective actions determined for the one or more challenges.

13. The computer-implemented method of claim 10, wherein the analyzing the sets of application activities to determine one or more challenges associated with the set of remote devices of the plurality of remote devices comprises:

determining that a first set of application activities of a first remote device of the plurality of remote devices is associated with a first challenge;

determining that a second set of application activities of a second remote device of the plurality of remote devices is associated with the first challenge; and

responsive to determining that the first remote device and the second remote device are both associated with the first challenge, generating the set of remote devices in association with the first challenge, wherein the set of remote devices include the first remote device and the second remote device.

14. The computer-implemented method of claim 10, further comprising transmitting the corrective action data structure to each remote device of the set of remote devices.

15. The computer-implemented method of claim 10, further comprising transmitting the corrective action data structure for the set of remote devices to a managing entity of the set of remote devices.

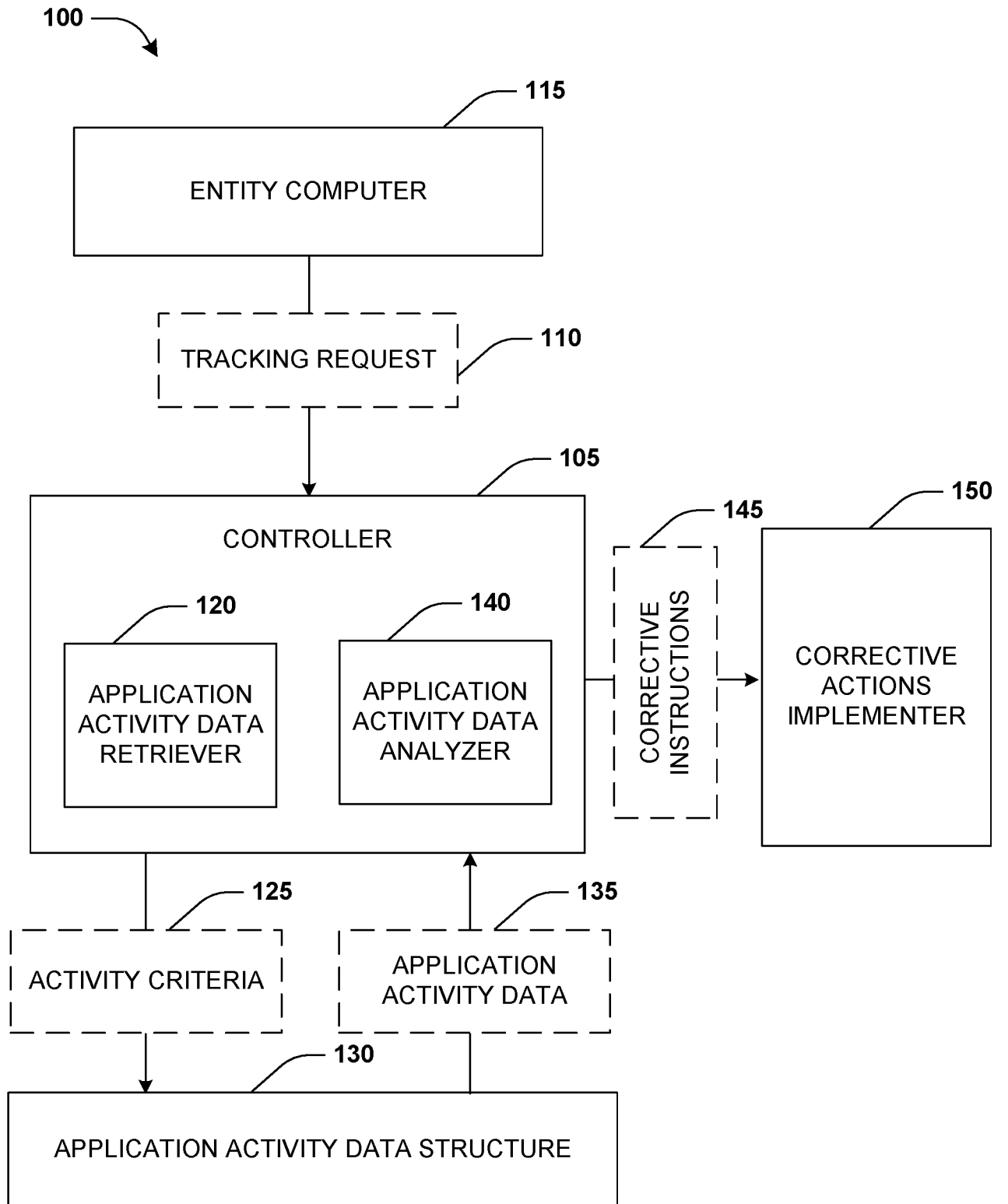
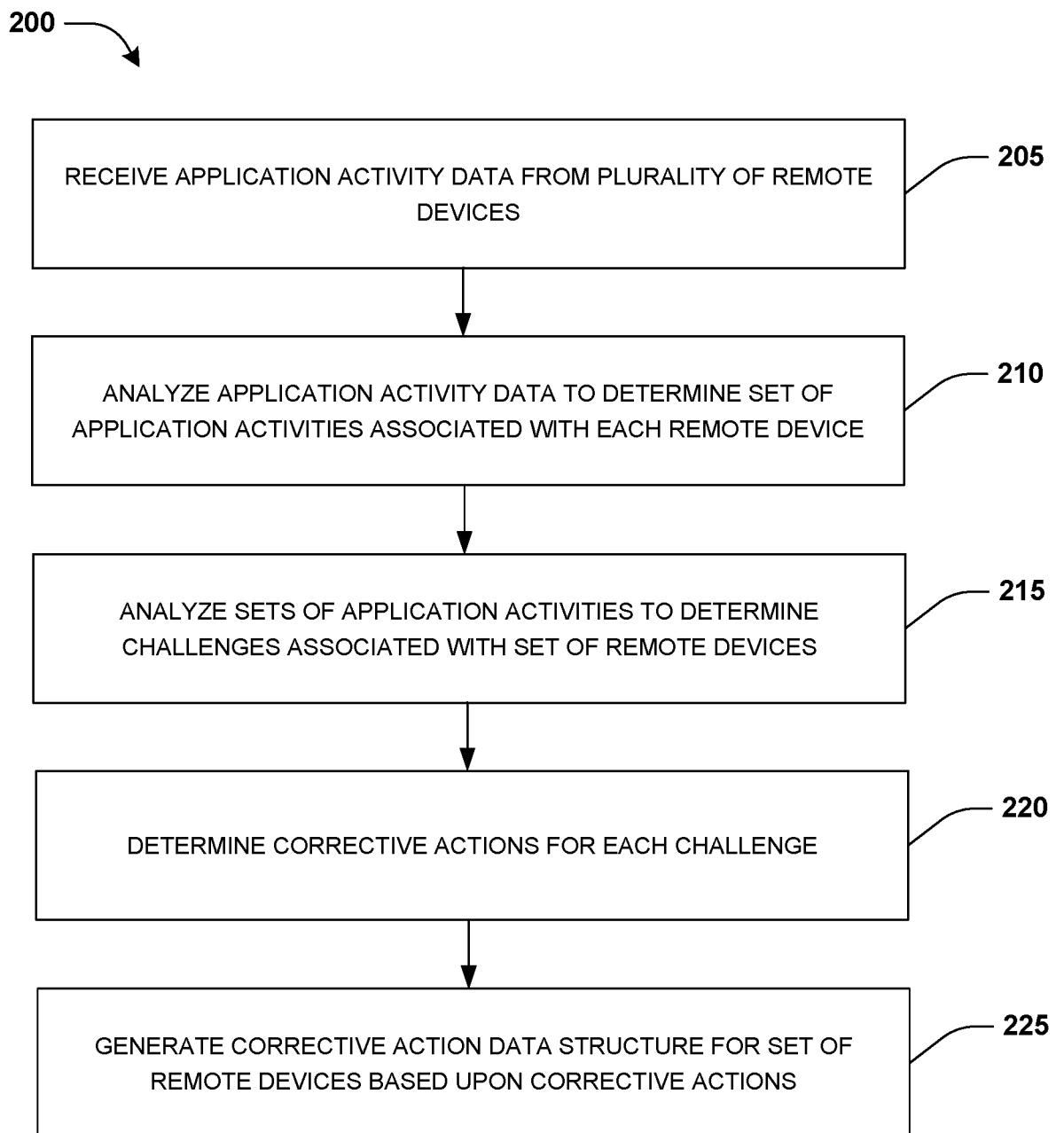


FIG. 1

**FIG. 2**

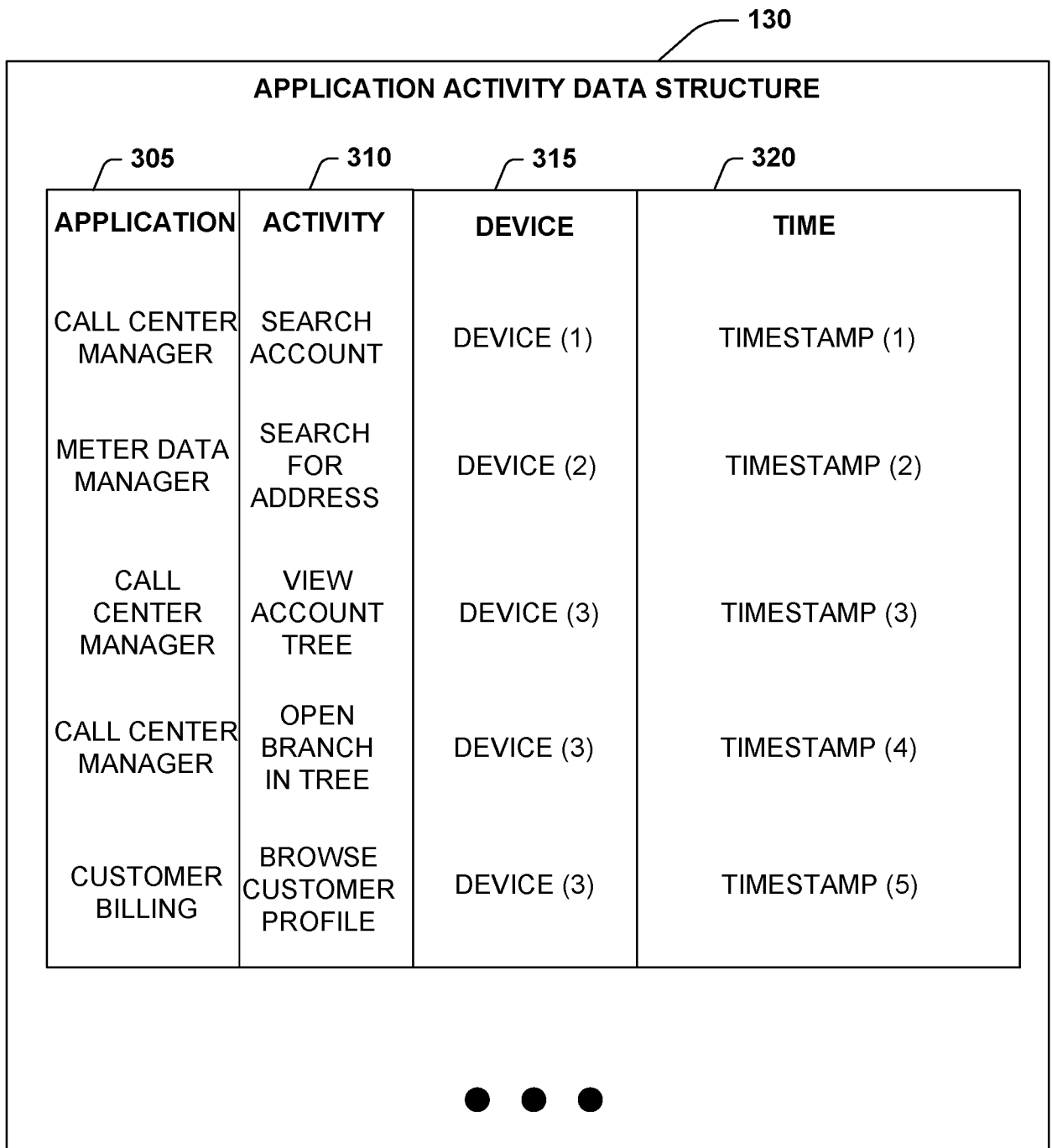


FIG. 3

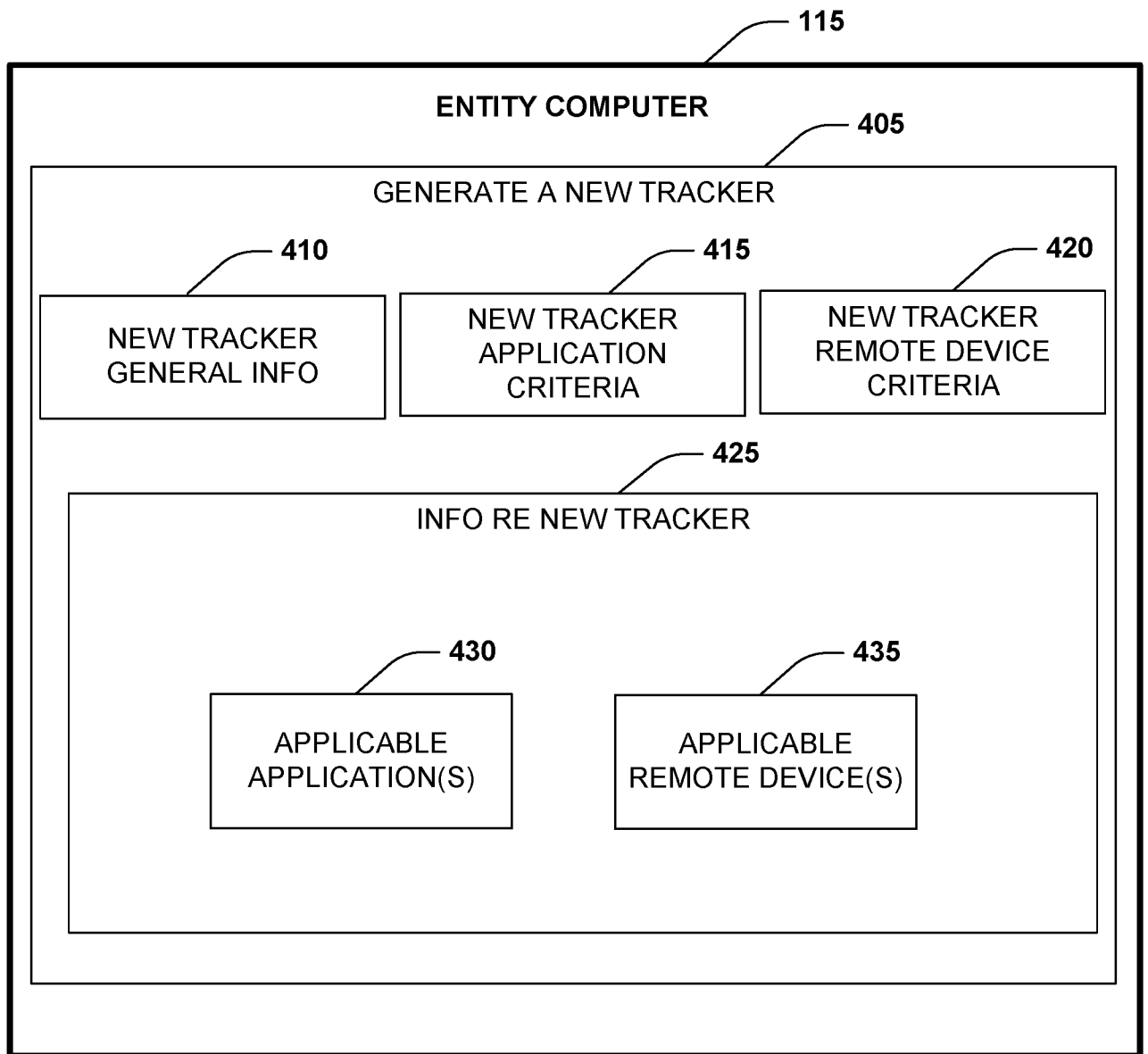


FIG. 4

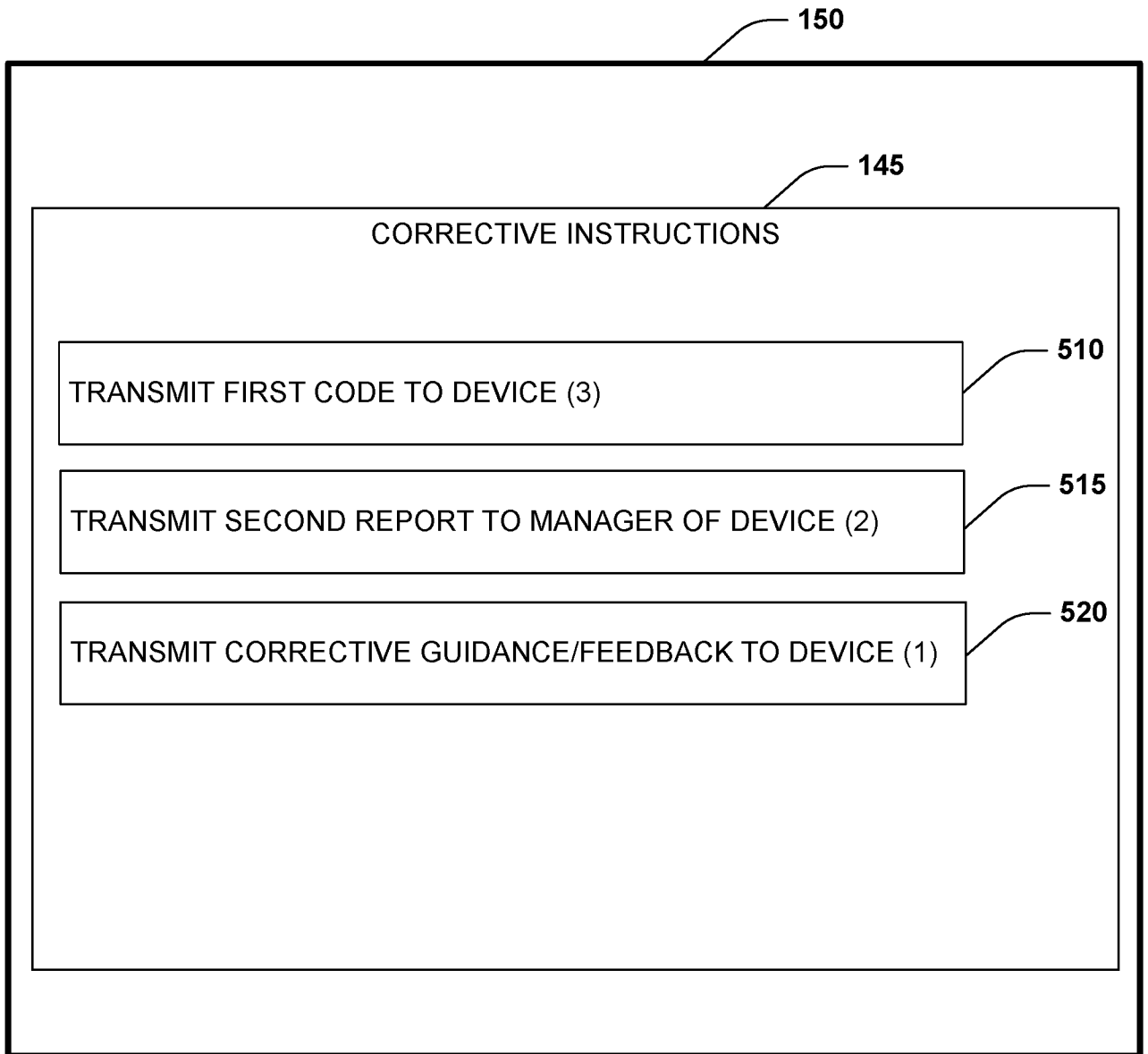


FIG. 5

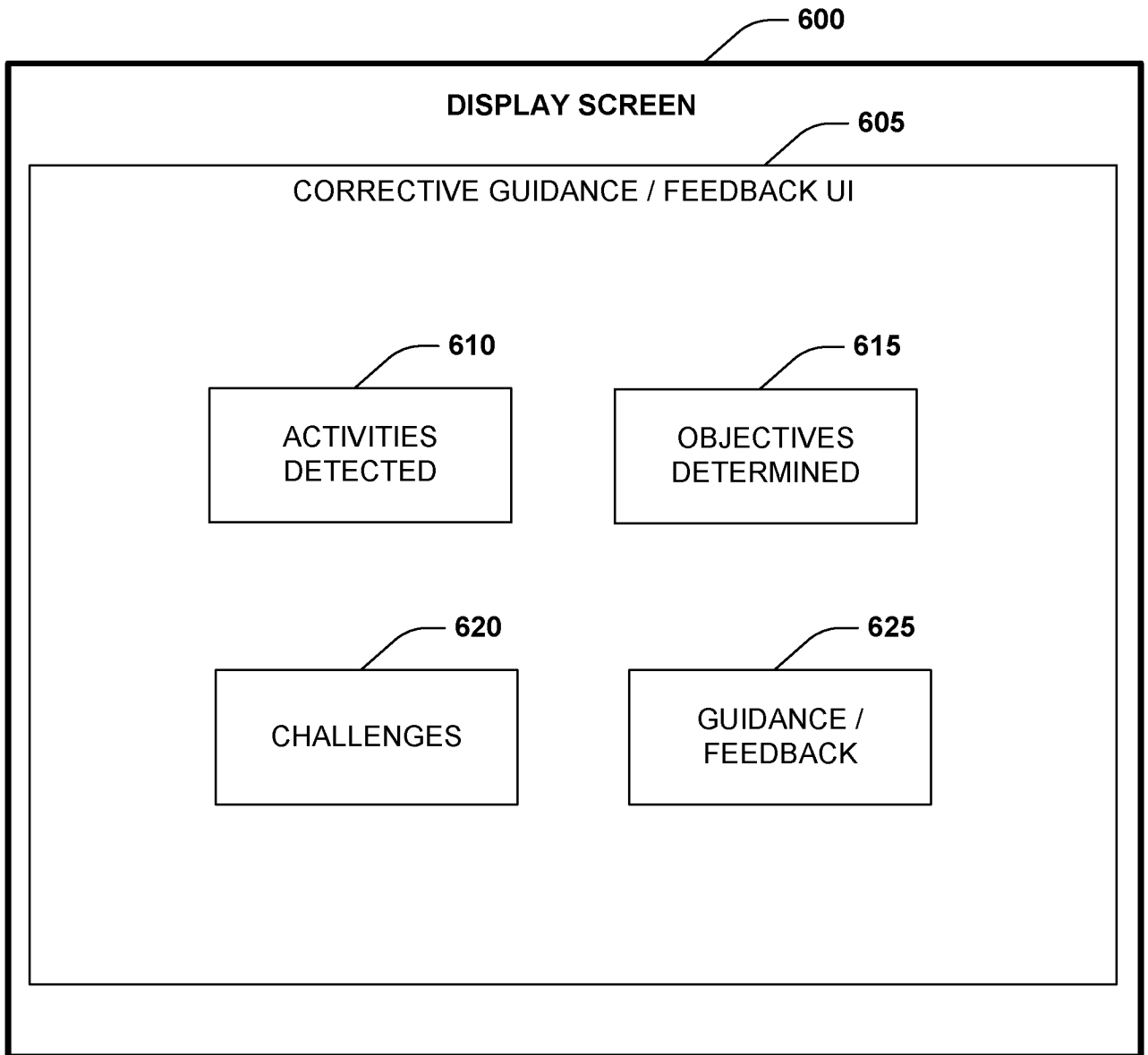


FIG. 6

700

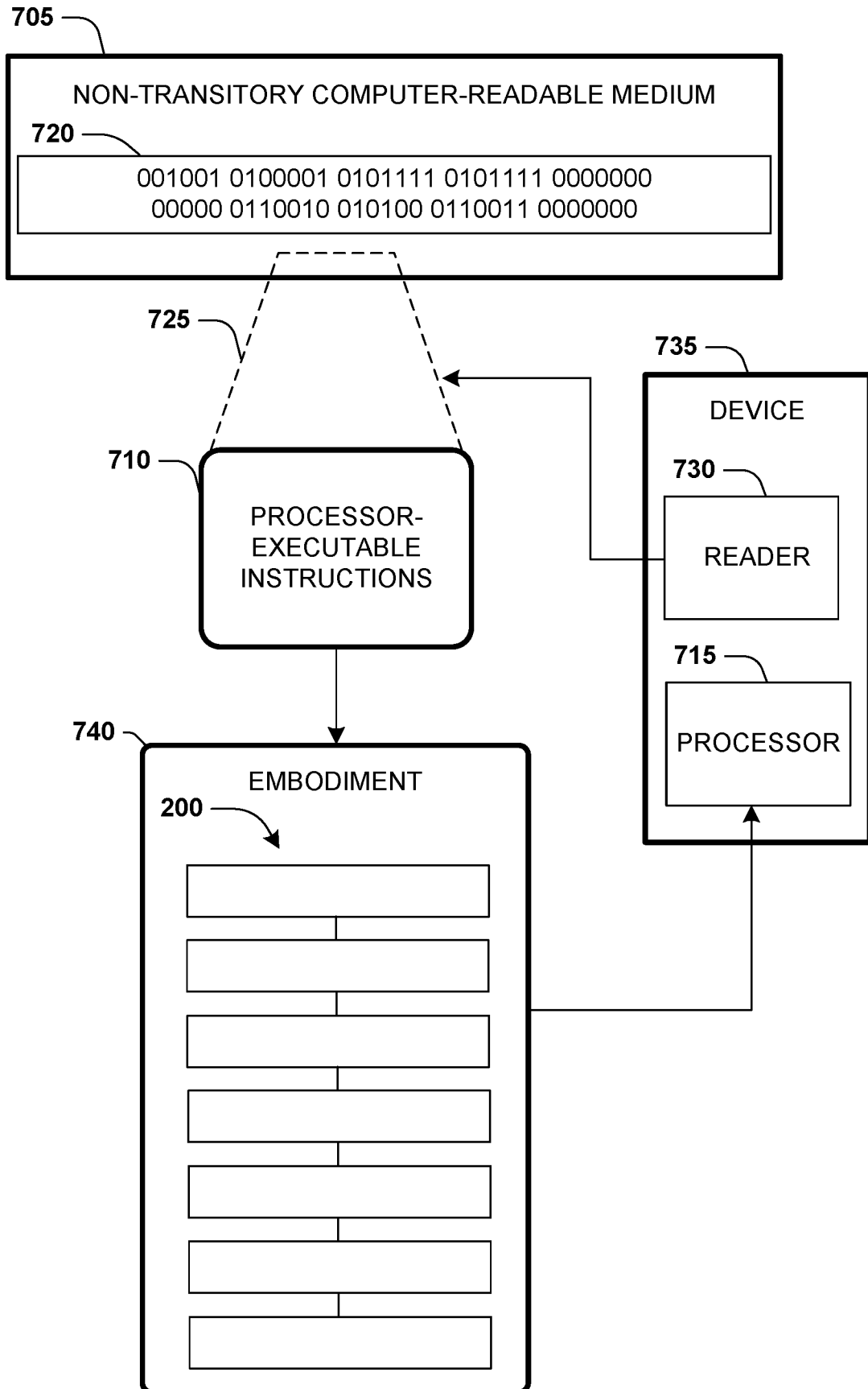


FIG. 7

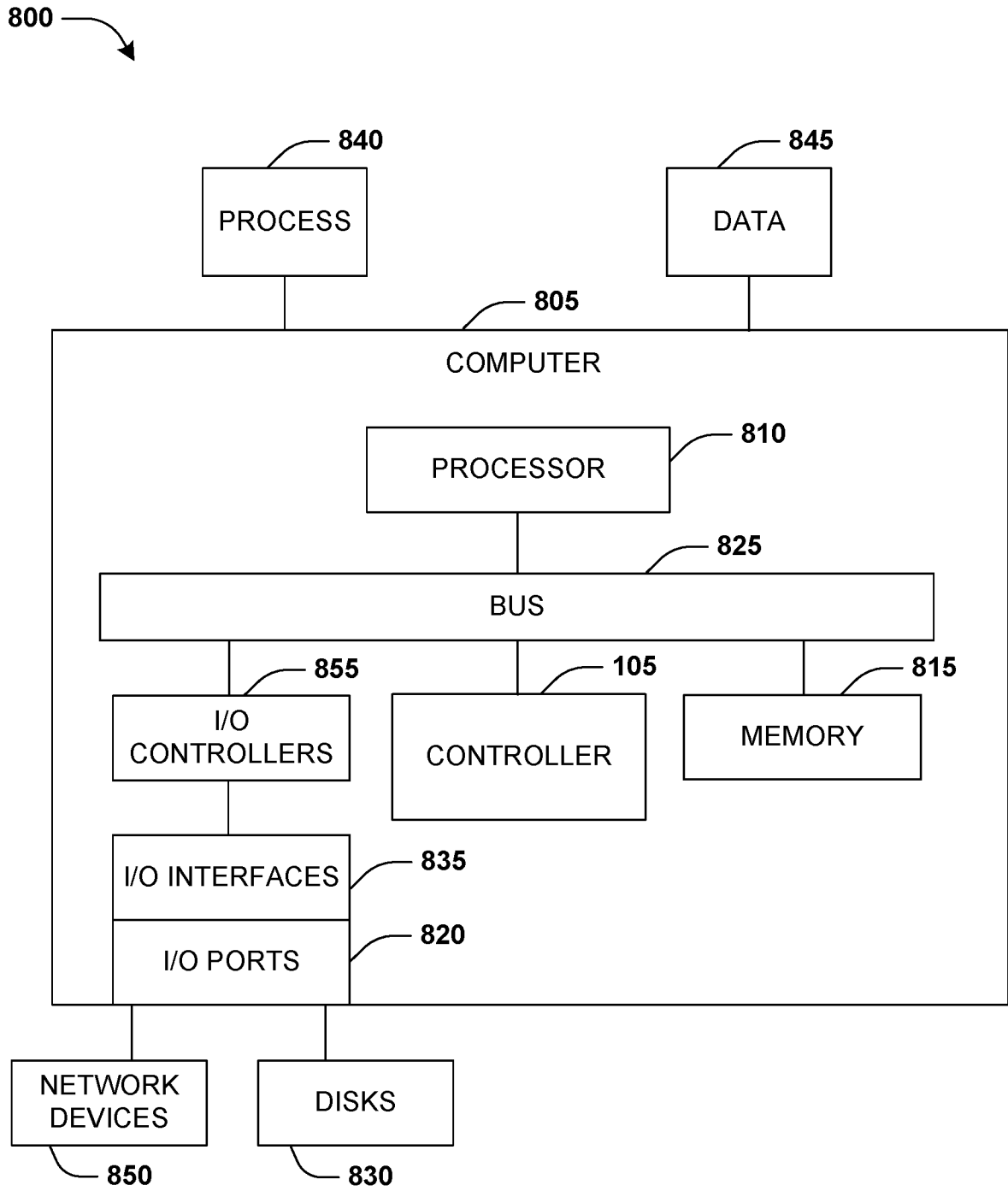


FIG. 8

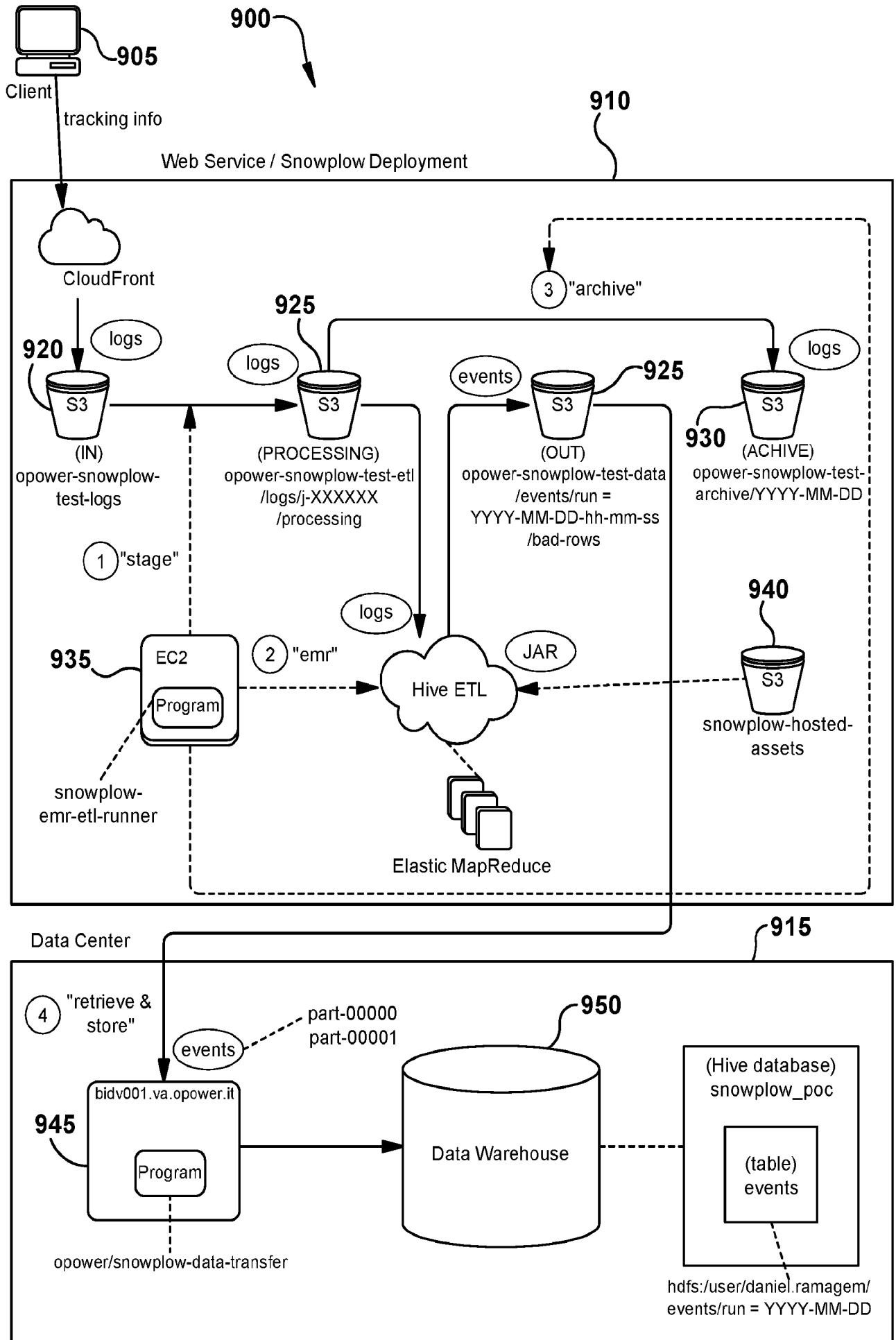


FIG. 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/064153

A. CLASSIFICATION OF SUBJECT MATTER
 INV. G06F9/50
 ADD. G06F11/34 G06F11/07

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 G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
 EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2015/067147 A1 (CARMEL IDO [US] ET AL) 5 March 2015 (2015-03-05) paragraph [0033] paragraph [0004] paragraph [0014] - paragraph [0023] -----	1-15
X	US 2006/229848 A1 (ARMSTRONG STEPHEN [US] ET AL) 12 October 2006 (2006-10-12) paragraph [0031] - paragraph [0033] paragraph [0041] paragraph [0053] -----	1-15

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search
 20 March 2019

Date of mailing of the international search report
 27/03/2019

Name and mailing address of the ISA/
 European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer
 Alecu, Mihail

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2018/064153

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015067147	A1	05-03-2015	GB 2518052 A
			US 2015067147 A1
			US 2016315837 A1

US 2006229848	A1	12-10-2006	CN 101278246 A
			EP 1929387 A2
			JP 5468261 B2
			JP 2009510601 A
			US 2006229848 A1
			WO 2007040785 A2
